Scientific paper awarding the academic title of

Master of Engineering

at “Technische Universität München”

Embedding food quality in simulation modeling for milk supply chains

Presented to: Prof. Dr. Martin Grunow

Chair for Production and Supply Chain Management

Technische Universität München

Supervisor: M. Sc. Bryndís Stefánsdóttir

Degree: Industrial Engineering

Author: Lorena Esteban Ponce
Matriculation Number: 03283610

Steinickeweg 4
80798 München

+49 (0) 176 32475916
lorenaestebanp@gmail.com

Date: October 4th, 2011

Signature Page

Declaration

This is to solemnly declare that I have written this paper all by myself. I did not use any additional sources than the mentioned ones. Ideas taken directly or indirectly from those sources are marked as such.

This paper or any of its parts have not been presented to this or any other board of examiners so far.

I am fully aware of the legal consequences of making a false declaration.

München, October 4th 2011

Lorena Esteban Ponce

Preface

"Embedding food quality in simulation modeling for milk supply chains" is the final project to finish the 5-year degree of Industrial Engineering, and will be acknowledged at my home university in Barcelona. I started this thesis in my 10th semester, which is also my second semester as an Erasmus student at the Technical University of Munich.

I am very fortunate to have had the possibility to write my thesis at the Chair for Production and Supply Chain Management at the TUM, and for that I want to thank my supervisor, Bryndís Stefánsdóttir, the head of the chair, Martin Grunow and the team of teachers and PhD Students working for him.

Many thanks to my parents and sister for giving me the opportunity to enjoy this personal and academic experience and for supporting me during the 5 years of my studies; and to all my friends for the support and company during the long working hours in the library.

Finally, thank you very much Lluis, for all the support and the help; for managing my temper and my nerves; and for cheering me up both from the distance and in person for the last few months.

Abstract

This thesis is part of a research study by the German Federal Ministry of Food, Agriculture and Consumer Protection (BMELV), aiming energy savings by producing milk and whey concentrates instead of milk powders, whose production process is highly energy intensive.

Although the new proposal is more sustainable, higher logistic efforts are likely to be necessary. The main objective of this study is to evaluate the trade-off between quality level of the product and logistic costs throughout the whole supply chain, and for that purpose, a simulation study has been implemented using the software Plant Simulation.

The current process for powders has been compared to 4 alternative processes for concentrates; which combined with two parameters (delivery frequency and cooling temperature) generate 16 different scenarios.

In order to design the simulation model, a top-down approach is used, allowing to independently model each of the processes involved, as well as to easily modify the model for more advanced stages of the bio-processing research. The simulation model is highly focused on individual batch quality, by means of quality prediction models, and batch traceability, both intrinsic to the model and its dynamic behavior (programmed by methods).

Finally, the simulation outcomes for each scenario, i.e. average product quality and total costs, have been compared to the powders reference scenario.

Table of contents

List of figures

List of tables

List of abbreviations

BBD	Best-Before Date
DC	Distribution Center
EC	Event Controller
FIFO	Firs In First Out
FSC	Food Supply Chain
FSCN	Food Supply Chain Network
KPI	Key Performance Indicator
MU	Moving Units
SC	Supply Chain
SCM	Supply Chain Management
SCN	Supply Chain Network
SCP	Supply Chain Planning
SEC	Specific Energy Consumption
T	Temperature
TC	Total Costs
TQ	Total Quality

1 Introduction

1.1 Problem definition

This thesis is part of a research study by the German Federal Ministry of Food, Agriculture and Consumer Protection (BMELV), aiming energy savings by producing milk and whey concentrates instead of milk powders.

The global project is motivated by the current trends in industry such as environmental friendly production processes, as well as energy and emissions savings; furthermore these have become environmental, commercial and economical priorities for all industry sectors (Kulozik and Grunow, 2011).

Additionally for the food industry, not only sustainability is expected but also high quality and safety requirements as well as product traceability; to the point that these are criteria affecting the consumer demand (Van der Vorst *et al.*, 2009).

Hence, the main objective of the research is to procure a competitive advantage for the German dairy industry by means of innovations in the process technologies of semi-finished products and evidence of their lastingness (Kulozik and Grunow, 2011).

Milk powders are a commonly used semi-finished product in the dairy industry, used by bakery, cheese and milk producers among others. The general simplified overview of the process starts with the transportation of the milk from the farm to the dairy plant, where it is dehydrated (water extraction) and afterwards sold as powder to the dairy producers, who will re-hydrate the powders (water addition) to obtain the final product.

The production process used in the current industry for powders, as explained in figure 1.1, has two different stages of water extraction: concentration by evaporation (alternatively reverse osmosis) and drying.

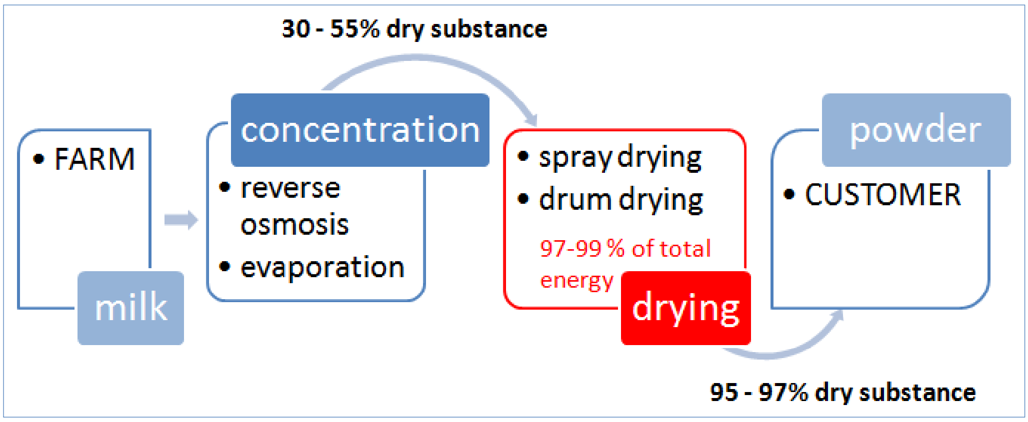

Figure 1.1 Current dairy production general process for milk powder

Thus, after the milking in the farms, the milk is transported to the dairy plant, then concentrated up to 55% dry matter and finally dried (spray drying process) before it can be sent to the customer. It is in this last processing step, the drying, where the semi-finished product with up to 97% dry matter can be obtained, nonetheless it requires almost 99% of the total energy used in the dairy process (Kulozik and Grunow,2011).

The basis of the new proposal consists in the elimination of the drying process, obtaining as final semi-finished product milk concentrate instead of milk powder, as shown in figure 1.2.

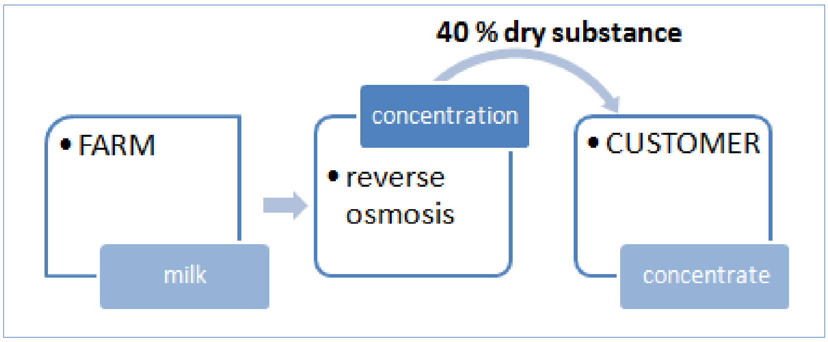

Figure 1.2 New proposal general process for milk concentrates

Hence, when delivered from the farm, the milk would be concentrated by reverse osmosis. As a result, concentrates would only have approximately 40% dry matter, fact that involves some positive and negative consequences.

Some of the main advantages of the concentrates are the lower energy consumption in the dairy process as intended, as well as the possibility to avoid the nowadays existing problem of clumping (grouping of the powder particles while storage or transportation) or re-dispersion (problems when mixing the powders with water again to obtain the final product) (Kulozik and Grunow, 2011).

On the other hand, the disadvantages are the two main consequences of higher water content. First, concentrates would have approximately a 2,5 times higher volume than powders, and consequently transportation and storage costs would increase. Furthermore, concentrates would have faster quality deterioration and therefore a shorter shelf-life.

As follows, the main points to cover within the different thesis in the study as described in the project proposal by Kulozik and Grunow (2011) are:

- Alternative processes for concentrates that allow these to be transported and storage at room temperature likewise powders are (up to 4 months).
- Energy savings and suitable methods to value those savings, especially when new means are necessary or when a conflict regarding the aim appears, for instance the conflict between mass reduction, lastingness and quality.
- Adequate logistics for concentrates, taking into consideration the additional energy and costs needed, as well as methods to value the extra resources, also regarding quality and safety requirements.

This thesis in particular is focused on the last point; that is the quality modeling of milk concentrates versus milk powders throughout the whole milk supply chain, depending on different process variations.

1.2 Research design

The research design can be divided in two main parts: first the study and analysis of the current milk powder supply chains, secondly the analysis and evaluation of the aimed new SC for concentrates.

As for the powders SC, the objective is to simulate a general system representing the standard conditions in Germany, under the assumption of a 'best practice' SC scenario, referring to a feasible SC configuration and operational management and control of all SC stages that achieves the best outcome for the whole system (Van der Vorst, 2000). This first simulation model is to be considered the basis scenario for the evaluation of the concentrates SCs and will be used as reference.

Accordingly, concentrate's SCs are to be simulated for different scenarios, these including key environmental and operational parameters, in order to evaluate the logistical and quality impacts of milk concentrates in reference to the basis scenario.

1.3 Outline

The reminder of the thesis is structured in six chapters as follows: in chapter 2 the state of the art will be described in addition to general background information on the most relevant investigation papers and recent publications on which the research is based. That includes literature on Food Supply Chains and the special requirements and conditions that should be considered in addition to general SCM; Simulation Environments used in other studies for FSC and the appropriate modeling of quality.

In chapter 3 the methodology used will be characterized, that is the reference scenario will be justified accordingly to the industry description, what will lead to the simulation model description. Finally, the model will be verified and validated in order to present the obtained results.

Furthermore, in chapter 4, the results will be analyzed to finally, come to the conclusions in chapter 5. Ideas for further research as well as the limitations of the study will be explained in chapter 6.

2 Review of literature and research

2.1 Food supply chain management

As mentioned in the introduction, consumer demand has become more demanding regarding food quality, integrity, safety, sustainability, diversity and associated information services in the past years (Van der Vorst *et al.*, 2009). This trend among consumers gains even more importance after recent accidents, likewise the E. coli crisis in Germany in May 2011.

According to Van der Vorst *et al.* (2005), the food industry is becoming an interconnected system with a large variety of complex relationships, reflected by the formation of FSCNs via alliances, horizontal and vertical cooperation, and forward and backward integration in the supply chain (see figure 2.1).

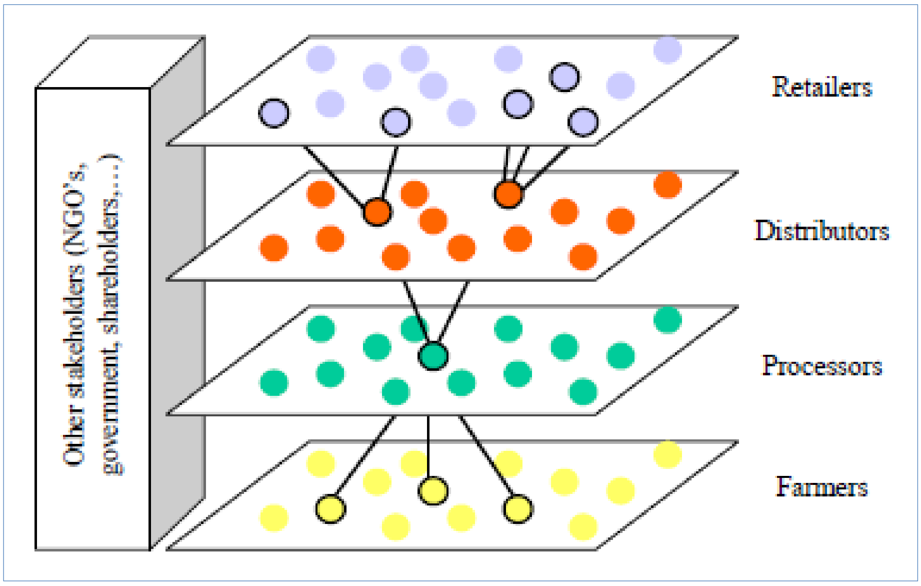

Figure 2.1 SC Diagram: processor's perspective (Van der Vorst et al. 2005)

FSC are also referred to by the term of agri-food supply chains (ASC), which refers to the activities from production to distribution with the objective to bring agricultural products from the farm to the table; moreover, these are formed by the organizations responsible for production (farmers), distribution, processing, and marketing of agricultural products to the final consumers (Aramyan et al., 2006).

Several authors and papers have recently focused on the peculiar features FSCs present with respect to other goods chains, for example Akkerman and van Donk (2007) consider following points the most important peculiarities:

- Limited time of storing due to limited shelf life and need for dedicated equipment and space
- ast processing by means of traced and high-quality systems and sequence dependent setup times.

Blackburn and Scudder (2009) show that conventional supply chain strategies are inappropriate for FSC because the main focus should relay in product value deterioration, which decays significantly over time in the SC and is highly temperature and humidity dependent.

Moreover, Van der Vorst *et al.* (2009) explain that equally important as the analysis of efficiency and responsiveness requirements is theanalysis of food quality change and environmental load of FSC.

As for the food quality change or food decay, the intrinsic focus on product quality makes the design of FSCs further complicated (Van der Vorst and

Beulens, 2002; Luning and Marcelis, 2006). For that reason, special attention is paid to this matter in chapter 2.2.

Regarding environmental load, sustainability in FSC focuses on the reduction of product waste, number of miles a product has travelled before it reaches the consumers' plate (food miles) and all greenhouse gas emissions related to the business processes in the SCN (carbon footprint) (Van Donselaar *et al.*, 2006).

Furthermore, Zanoni and Zavanella (2011) consider energy a key element within FSCs, due to the fact that it is necessary to guarantee quality-based processes. Moreover, they explain that the use of energy implies the consumption of resources, which directly affects the FSC's performances, including sustainability and economical.

Van der Vorst (2000) organizes some of the main particular requirements of FSC by Van Rijn and Schijns (1993), Rutten (1995) and Den Ouden et al. (1996) in table 2.1 and categorizes them according to the Supply Chain stage involved.

As Van der Vorst *et al.* (2009) conclude, further research in FSC should focus on improving the logistics performance in addition to the environmental sustainability and food quality preservation.

Table 2.1 Overview of the main FSCs characteristics (Van der Vorst, 2000)

SC stage	Product and process characteristics
Overall	▪ Shelf life and quality decay for raw materials, intermediates and finished products throughout the SC ▪ Recycling of materials required
Growers Producers	▪ Long production throughput times ▪ Seasonality in production
Auctions Retailers	▪ Variability of quality and quantity of supply (farm-based inputs) ▪ Seasonal supply of products requires global (yearly) sourcing ▪ Conditioned transportation and storage means
Food industry	▪ High volume, low variety production systems ▪ Focus on capacity utilization due to the highly sophisticated capital-intensive machinery ▪ Variable process yield in quantity and quality due to biological variations ▪ Possible delay in planned production because of quality controls ▪ Alternative installations and recipes, and cleaning and processing times depending on the product ▪ Necessity for lot traceability of work in process due to quality and environmental requirements and product responsibility ▪ Limited storage capacity when raw material and/or products need to be kept in special facilities

2.2 Modeling of quality

As mentioned in the previous section, food quality is of extremely importance for FSC and therefore a very relevant characteristic in plenty of research papers.

Due to the importance of product quality in the food industry, Trienekens and Zuurbier (2008) expect quality assurance to dominate production and distribution processes, in addition to the increasing efficiency and cost reduction motivated by the costs for certification, auditing and quality assurance (Akkerman et al., 2010).

Food quality is not only a performance measure, but also directly related to other food attributes like integrity and safety (Van der Vorst *et al.*, 2009).

In order to quantify the product's quality level and as explained by Grunert (2005),it should be taken into consideration the fact that food quality usually refers to the physical properties of food products, as well as to the product perception by the final customer, which can include microbial aspects, texture or flavor among others.

Nevertheless, Rong *et al.,* (2010) consider that regarding the wide range of product characteristics, most quality prediction models use one leading quality characteristic for each given product.

Especially in fresh food products, food quality is determined by biological variations, in addition to time and environmental conditions, i.e. temperature, humidity and presence of contaminants, all factors that can be influenced by following characteristics, according to Van der Vorst *et al.*, (2009):

- Packaging material
- Loading processes
- Temperature conditioned transportation means and warehouses

Some authors focus on the modeling of food quality change by using time temperature indicators (TTI) in order to trace the temperature conditions of each product batch individually throughout distribution (Taoukis and Labuza, 1999; Schouten *et al.*, 2002; Tijskens, 2004).

Obviously, temperature is an important factor in controlling product quality in supply chains. The rate of quality degradation k is therefore often based on the Arrhenius equation, a formula for the temperature dependence of a chemical reaction. The general form of this equation is:

$$q = k_0 \cdot e^{-(E_a/RT)}$$

where k_0 is a constant, E_a the activation energy (an empirical parameter characterizing the exponential temperature dependence), R the gas constant, and T the absolute temperature (Rong *et al.,* 2010).

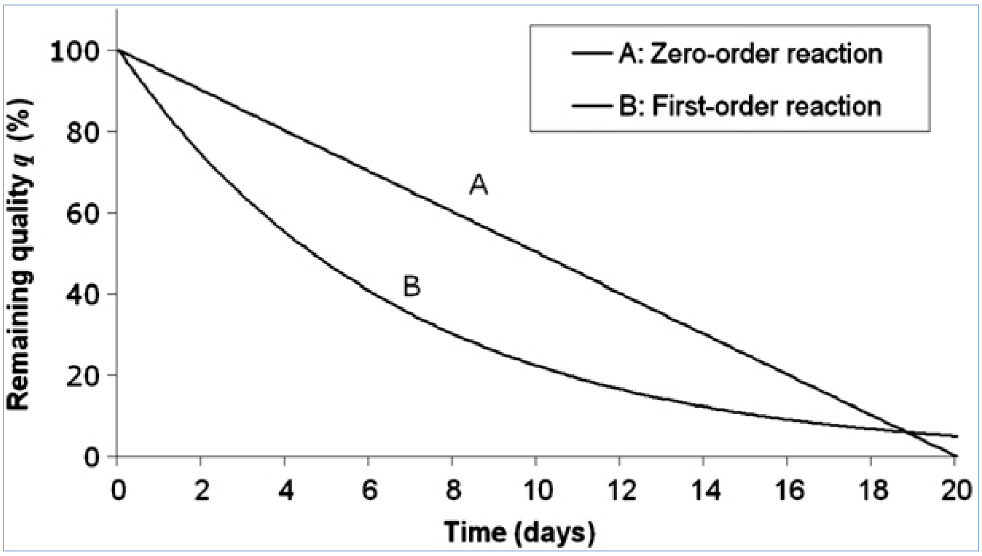

Figure 2.2 Illustration of quality degradation of food products (Rong et al., *2010)*

The above presented equation leads to the quantification of the quality change during a specified time period for each possible storage or transportation temperature (Rong *et al.,* 2010):

$$q = q_0 \cdot \exp[\, k_0 \cdot t \cdot e^{-(E_a/RT)}]$$

Accurate shelf-life prediction is an important aspect of food science, not only to corporations but to governments and the general public as well. A premature loss of shelf-life can lead to a loss of consumer confidence and of revenues to the food manufacturer. Shelf-life testing also allows the company to minimize costs in formulation and packaging. (Fu and Labuza, 1993).

Moreover, in the recent years, next to high quality levels, the need for an accurate chain control and its monitoring has emerged as one of the most critical issues (Montanari, 2008). The precautionary principle in the General Food Law requires food business operators to ensure food safety in the food chain (EC Regulation, 2002), that is traceability of food products has become a crucial matter. According to ISO Quality Standards, traceability is defined as: “the ability to trace the history, application or location of an entity by means of recorded information” (ISO 8402:1994).

In the food chain, traceability means the ability to trace and follow a food, feed, food-producing animal or substance through all stages of production and distribution (Manikas and Manos, 2008).

2.3 Simulation environments for FSC

The VDI (Association of German Engineers) guideline 3633 defines simulation as the emulation of a system, including its dynamic processes, in a model one can experiment with. It aims at achieving results that can be transferred to a real world installation and at defining the preparation, execution and evaluation of experiments within a simulation model.

According to Huang *et al.* (2003), discrete event simulation is a natural approach for supporting supply chain network design, due to the difficulty to perform an analytic evaluation because of their complexity. Nevertheless, discrete event simulations tend to stress logistics analysis rather than product quality or sustainability (Van der Vorst *et al.*, 2009).

Usually, SCs are cost or service driven but, recently environmental considerations in the SCP models are gaining importance by the addition of environmental constraints to (Subramanian et al., 2010), by developing multi-objective functions including profitability and sustainability (Quariguasi et al., 2008), or by using simulation to evaluate trade-offs between environmental and economic performance (Akkerman and van Donk, 2010).

Especially for FSC Simulation, time and temperatures become two essential factors. For exemple, Van Donselaar *et al.*(2006) use time-dependent quality information in the design of perishable inventory management systems. Moreover, Zanoni and Zavanella (2011) explain the need to model the chain itself for the optimization of the FSC, taking into consideration the temperature set and its impact on quality, energy and associated costs, referring to the fact that the lower the temperature, the higher the energy required and the longer the product life.

In addition, high demands are set on model transparency and completeness. Transparency refers to the insight into model components and their workings, whereas completeness addresses a full overview of design parameters (Van der Vorst *et al.*, 2009). This leads to following requirements on simulation model design according to Van der Zee and Van der Vorst (2005):

- Model elements and relationships: Hierarchy and coordination are important decision variables, which require an explicit definition of actors, roles, control policies, processes and flows in the model. Therefore agents, jobs and flows can be used.

- Model dynamics: stock levels and lead times are an important issue given the many parties involved; therefore timing and execution of decision activities should be explicit. This requires the ability to determine the dynamic system state and calculate the values of multiple performance indicators at all times. As Van der Vorst *et al.* (1999) remark, the model should be able to calculate the state, time and place of each business entity after each transition.

  This can be realized by the job execution, which can be triggered by multiple causes and have processing times depending on the entities processed and process capacity.

- User interface: The need for the chain partners to get involved in the simulation study is required for two reasons: to consider the solution trust-worthy and a better acceptance of the study's outcomes; and secondly to achieve a better solution in terms of model correctness and quality of the scenarios (McHaney and Cronan, 1998; Bell *et al.*,1999; Robinson, 2002). Therefore, an explicit representation of decision variables leads to visibility and better understanding of all processes in the model. The authors suggest the use of basic logistic terminology and recognizable building blocks.

- Ease of modeling scenarios: Given the complexity of the SC and the large number of possible scenarios, the choice of building blocks, the time required to adapt them to model specific requirements and the possibility to reuse models should be taken into consideration in order to increase the speed of modeling and analyzing alternative scenarios.

Also Beamon and Chen (2001) and Kleijnen and Smits (2003) introduce the need for the model to allow for the tradeoff between logistical costs, service, and product quality indicators. Moreover, they explain that the agreement on a FSCN scenario is reached based on the evaluation of the consequences of *KPI* (defined by Fortuin (1988) as variables indicating the effectiveness and/or efficiency of a part of or the whole of the processes or systems compared with to a given norm/target or plan), given the restrictions set by the available resources.

Whereas traditional performance measurement systems are based on costing and accounting systems, the special characteristics of FSCs require a more balanced set of economic and operational measures (Lohman et al. 2004). The choice of KPI should include investment and operational costs, as well as customer service, that is on-time delivery and product quality.

Some of the main ideas used as reference for this thesis can be found in different case studies by several authors:

Van der Vorst *et al.*, (2008) use following key performance indicators to measure effectiveness and efficiency of alternative designs:

- Distribution costs, including transport and warehousing only.
- Energy and emissions during distribution.
- Product quality when arriving at the retail store, measured by the remaining number of days until the predetermined BBD (remaining selling time), the remaining keepability of the product at a storage temperature of 4ºC and the percentage of products for which the BBD is not reached yet, but the quality state is no longer acceptable.

Van der Vorst *et al.*, (1999) use business entities to represent an information flow and / or goods flow. For that purpose, each business entity has a unique identification, a time stamp (keep track of the connection between input and processed entities for tracking, tracing and performance measurement) as well as descriptive attributes.

For the case study by Wang *et al.*, (2011), the determination of the delivery frequency is essential: it will affect the transportation cost and the quality deterioration. The simulated scenarios consist on the combination of two possible temperatures for chilling and two different packaging materials.

3 Methodology

In order to fulfill the assignments of this thesis, the working methodology comprises several stages, these including research and familiarization with the industry, literature research on FSC, quality simulation and simulation environments, as well as the familiarization with the software Plant Simulation and the learning of the specific programming language SimTalk.

The overall simulation methodology starts with a first impression of the real-world installation, followed by some information and data collection for the creation of the new model. These are then abstracted to become a simulation model according to the aims of the simulation studies, followed by the interpretation of the data produced in the simulation run (Tecnomatix Plant Simulation Help).

3.1 Top down approach

After the initial background research on the general project and the objectives to be covered within this study, and in order to guarantee model transparency and completeness, a top down design is chosen.

For this purpose, the first model, that is the top layer, consists of a general milk supply chain only including the farms, the dairy plant, the distribution center (DC) and two different customers, as well as the transportation between parties. (See figure 3.1).

Each of the SC stages is modeled by a different frame within Plant Simulation, fact that allows to subsequently modeling the different processes in each stage with so many details as necessary.

Afterwards, all the processes involved are modeled in the corresponding frames, building the second layer. The connections between frames are modeled by interfaces.

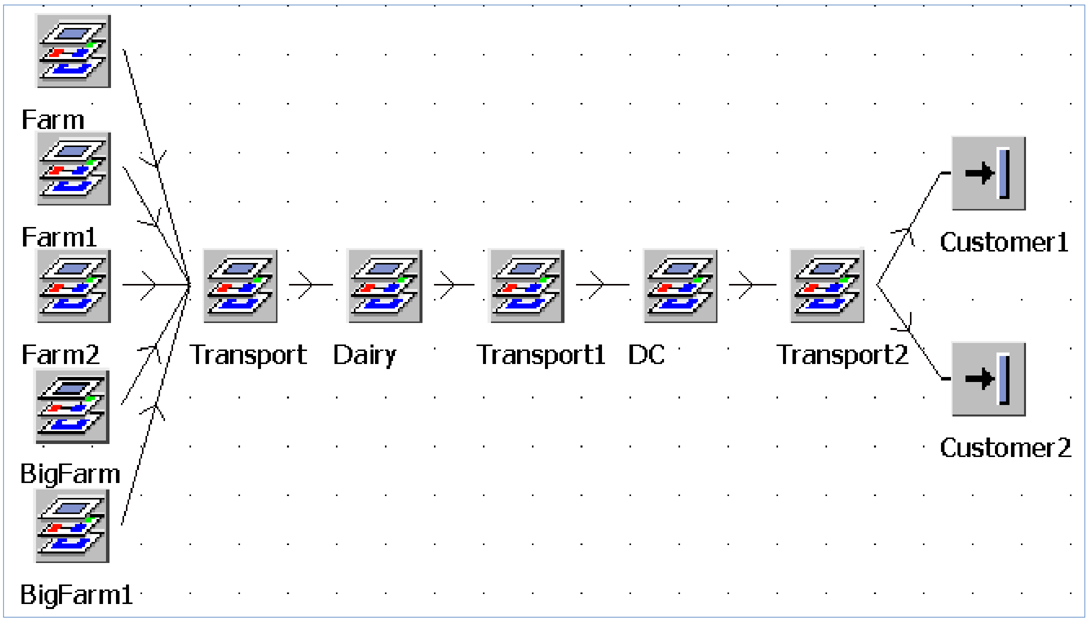

Figure 3.1 Top layer of the simulation model: CHAIN

Some of the frames contain more simply-modeled processes, which are completely implemented in the second layer, as in the case of the farms; other frames contain more complex structures.

In these cases, sub-frames are built, as for example the transport frame, which contains three different sub-frames (building the third and last layer):

loading process, transportation and unloading process; as shown in figure 3.2.

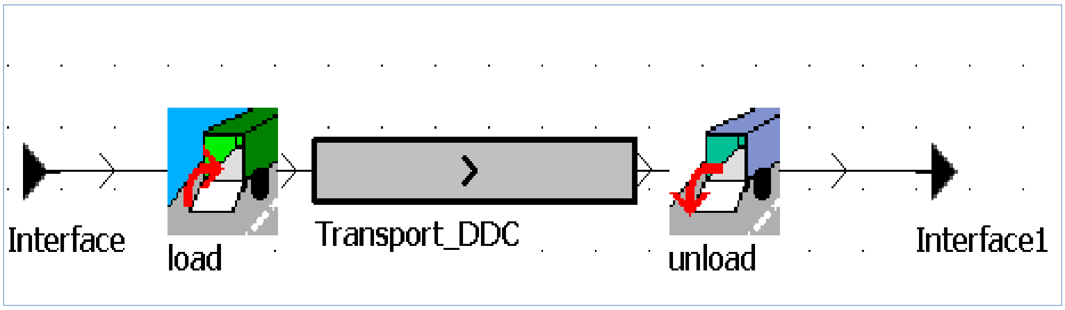

Figure 3.2 Second layer of the simulation model: Transport

3.2 Use of frames and operational blocks

Furthermore, working with different frames provides the possibility to use these as blocks, which can be used repeatedly; for instance in the case of the transport frame (first layer) or the quality control frame, used in the dairy plant and the DC (second layer).

The same idea, creating a new operational block with the basic options of the software, is also used for the main elements of the process. In other words, most of the processing and storage steps, or even some stages in the SC, have certain common attributes and characteristics. These are modeled in a generic operational block (group of several basic Plant Simulation structures or objects), which can be later modified to fit the specific needs of the represented stage by only changing the parameters, avoiding the implementation of the whole block from the beginning every time.

Both, frames and operational blocks, are implemented in the class library of the model, and can be used when needed by adding these to the model.

Some of the advantages of this procedure are the high flexibility provided to the model, since it is easier to partly modify the simulation model by adding or deleting operational blocks; as well as the possibility to built an operational block and test it for correct computation (verification) instead of testing every object in the model.

3.3 Model dynamics implementation

In order to properly model the simulation dynamics, chiefly information flows such as process and storage timings, lead times, stock levels, etc. the option *Method* provided in Plan Simulation is used.

The object *Method* is executed during the simulation run and allows to program controls which will affect other objects' behaviors. In this study, a *Method* can be triggered in three different circumstances:

- In the beginning of a process: The method is in this case an entrance control and starts executing every time a part is starting a process.
- At the end of a process: The method is in this case an exit control and starts executing every time a part is leaving a process.
- Continuous run: The method is triggered by the *init method*, that is, every time the simulation starts (play button), and it runs during the total simulation time, until the simulation is stopped (stop button).

The general structure of the *methods* is explained in figure 3.3:

```
is
  declaration of local variables
do
  commands/orders
end
```

Figure 3.3 Structure of the Method

Regarding the implementation of the different methods, two different methodologies were used, once for continuous running methods and a different one for entrance and exit controls.

Regarding the continuous run methods, these are implemented in the frame were they are to be executed. In order to address the different process steps, global variables, parts, etc. absolute paths were needed, that is the location of the object was described by the names of all the frames involved beginning from the class library.

For instance, to address the loading process in the transportation from the Farm to the Dairy, following path was needed:

.classLibrary.Frame.subFrame.object → .powders.CHAIN.Transport.load

As for the entrance and exit controls, they are also built in the class library and used in a similar way than the operational blocks. In this case, anonymous identifiers are used to address the objects, fact that provides the possibility of a more flexible and simpler programming.

Table 3.1 Anonymous identifiers

Identifier	**Addressed object**
@	designates the part (entity) that triggered the control
?	designates the process that triggered the control
~	returns the *frame* within which the *Method* object is located
root	designates the topmost *frame* in the hierarchy of *frame*s
self	designates the currently executed method

3.4 Factorial design

With the simulation results, a full factorial design will be conducted for the quality KPI having as two level factors the cooling (low: cooling, high: room temperature) and the delivery frequency from the farm (low: 1day, high: 2 days) for each possible production process.

The statistical analysis will be performed by a free version of the software MiniTab and should serve only as a possibility to analyze the simulation outcome, meaning, the results themselves do not have much value, since the process are still in research and most of the real data is still missing

4 Simulation study

4.1 Scope

In this chapter the complete model, as well as the simulation study are described in detail.

For this purpose, the model is justified in section 4. 2 with an industry description in addition to an extended overview of the main processes (4.2.1), moreover the specific objectives, as well as the definition of the key performance parameters (KPI) and the selected scenario parameters are defined in section 4.2.2. Furthermore, overall model assumptions are presented in section 4.2.3 to conclude with the presentation of the several existing numerical quality models for liquid milk, powders and concentrates as well as the justification for the selected models and assumptions (4.2.4).

In section 4.3, the complete model is to be described: first of all, the material flow objects (4.3.1) and the mobile units (4.3.2) used in the model will be introduced together with their attributes and properties, as well as how the quality methods are implemented (4.3.3); to continue with the top layer, the chain (4.3.4), which is common for powders and concentrates; followed by the second-layer frames in detail: farms (4.3.5), transport from the farm to the dairy plant (4.3.6), dairy production processes (4.3.7), transport from the dairy to the DC (4.3.8), DC (4.3.9) and finally transport from the DC to the customer (4.3.10).

Regarding the 6 above mentioned sub-sections, firstly, the stage for powders is described and in second place the possibilities and variations for each concentrate production process will be discussed. Additionally, global variables and their calculations will be justified in section 4.3.11.

The remaining section of chapter 4 is dedicated to the model verification (4.4.1) and the model validation (4.4.2).

4.2 Modeling

4.2.1 Industry description

A very simplified industry description of the dairy processing, regarding the part dedicated to milk powders, and the description of the concentrates research project is given respectively in figures 4.1 and 4.2.

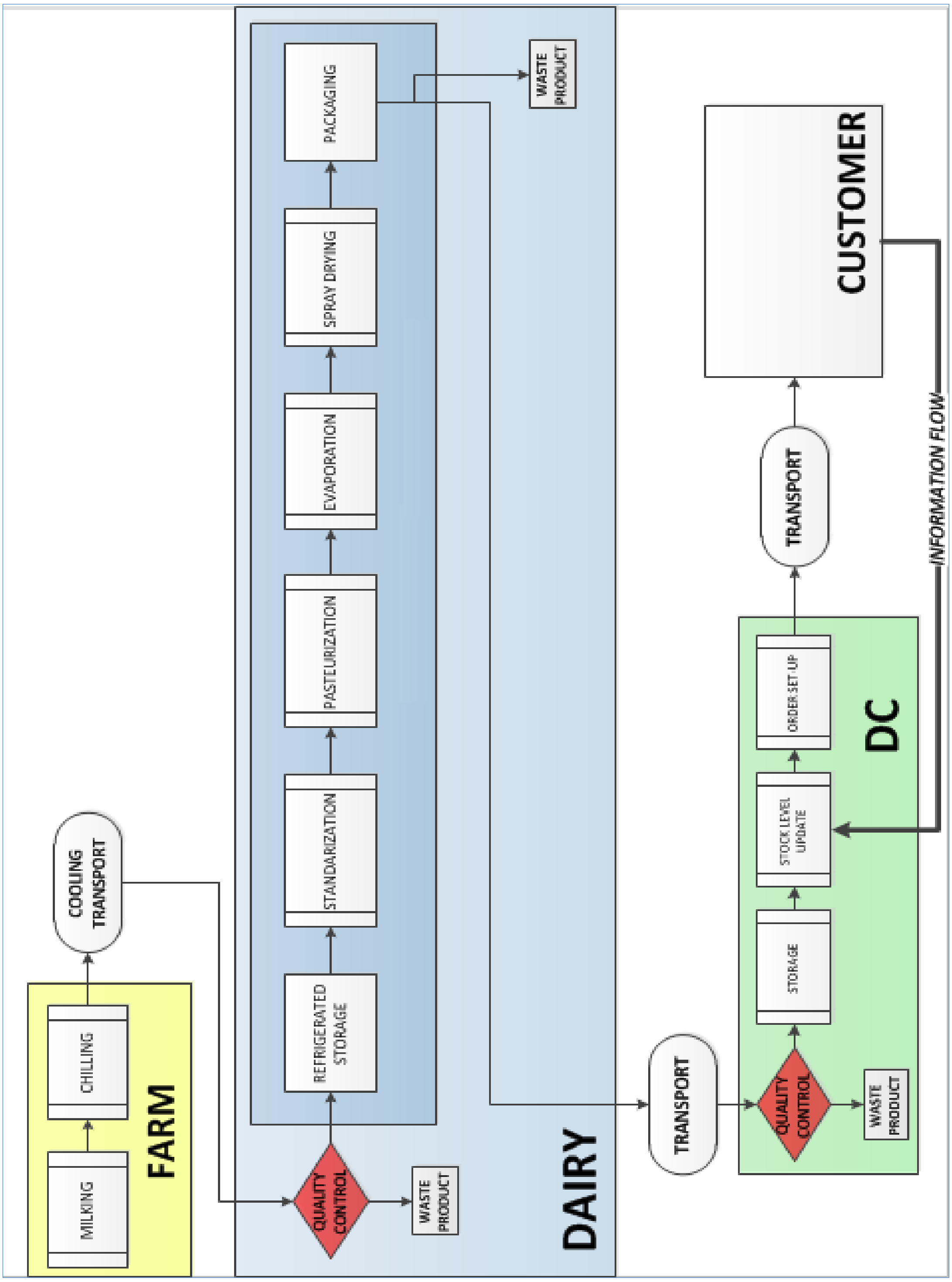

Figure 4.1 Schematic overview of the powders production process

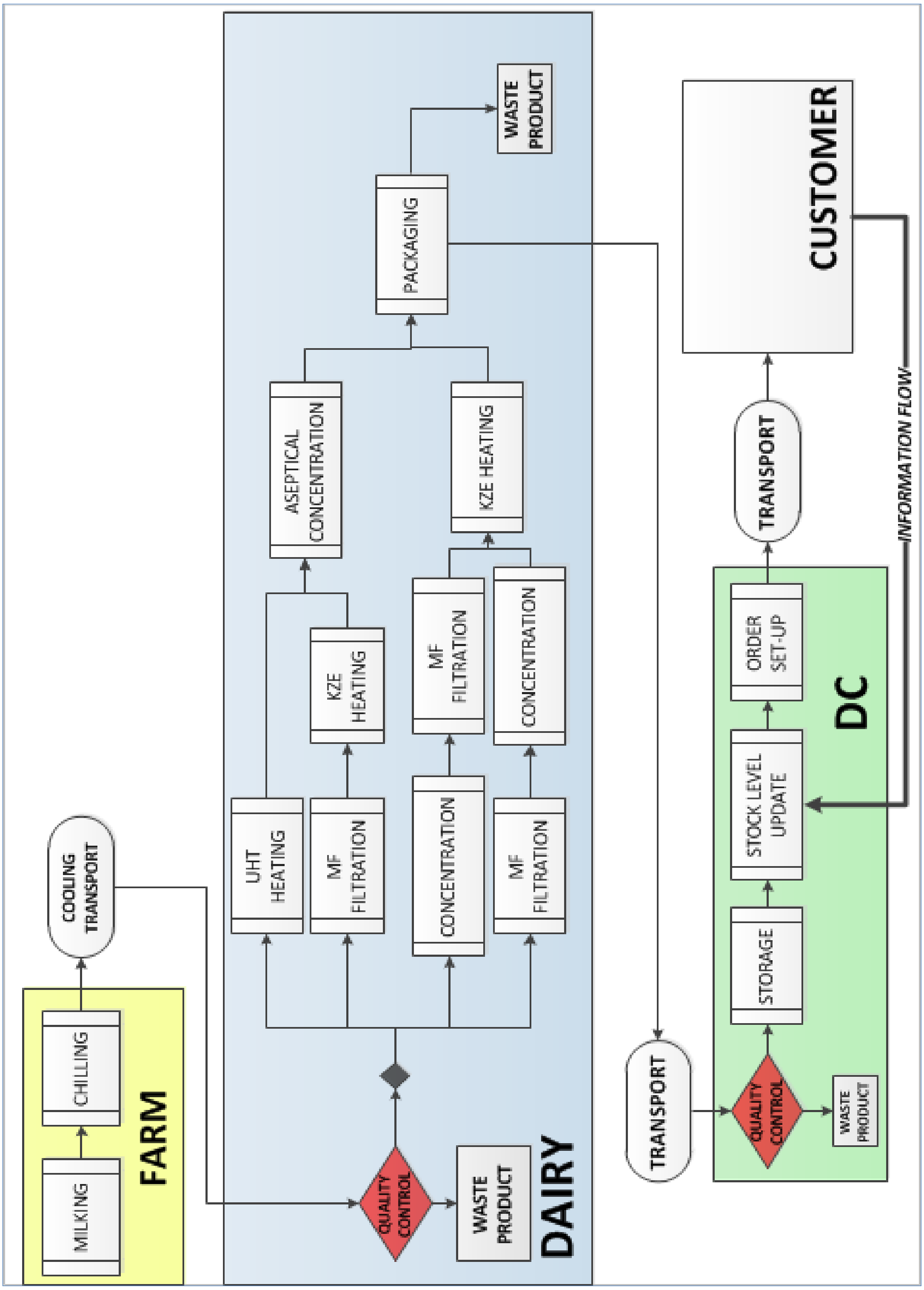

Figure 4.2 Schematic overview of the concentrates production process

4.2.2 Objectives, scenarios and key performance parameters

The specific objectives of the simulation modeling are to contrast and compare the advantages and disadvantages of the possible new production processes for milk concentrates, both between themselves and in comparison with the actual conditions of milk powders; as well as the quantification of these. In order to do so, different scenarios will be modeled.

The outcome of the simulation should serve as a first evaluation of the proposal feasibility, and if so, it should provide some directions on which of the concentrate production process better fits the industry needs.

Regarding the early stage of the project, the simulation is also to be considered a basis to work on, meaning it should be adapted to the concrete chain characteristics to achieve more accurate results on more advanced stages.

Turning to the scenario definition, the simulation study involves five different situations: the fist scenario (S0) intends to illustrate the average German SC for powders, describing the industry and its processes the way they currently are. S0 is to be considered the reference scenario and will be the basis to evaluate scenarios S1 to S4. Also the comparison between these will be based on S0.

Hence, scenarios S1 to S4 will represent the four alternative processes to produce milk concentrates which are being studied by the department of food engineering at the Technical University Munich.

Firstly we can distinguish between processes having the heating and/or the filtration process in the first stage followed by the concentration process in the second stage (S1 and S2). On the contrary, in the first stage of S3 and S4, the concentration and the filtration processes take place, followed by the heating process, as shown in Figure 4.3.

Thus, the specific order of the necessary processes will affect the conditions under which the milk is to be processed in the remaining stages, in addition to accordingly temperatures and processing time requirements. The exact characteristics of each process will be described in section 4.3.5.

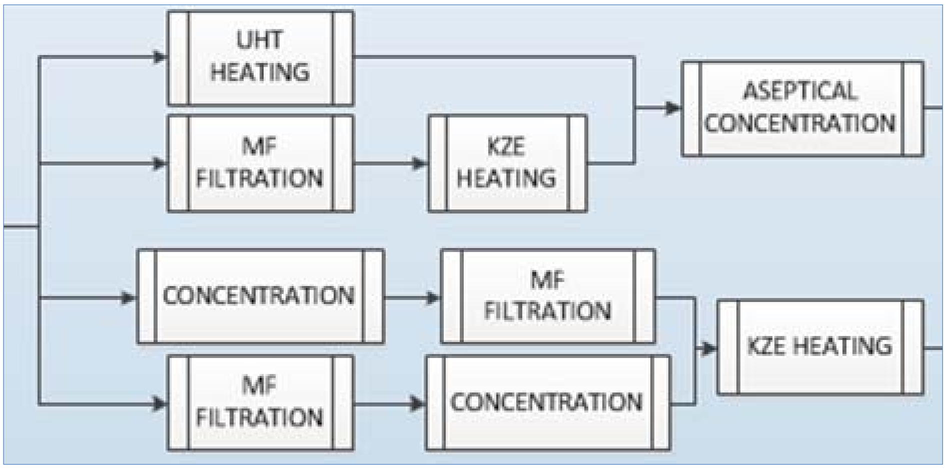

Figure 4.3 Process diagram for scenarios S2 to S4

Hence, to quantitatively measure the differences between scenarios, two key performance indicators (KPI) will be taken into account: total cost (TC) of the supply chain and average quality (Q) of the semi-finished product (powders or concentrates) when delivered to the customer.

As for the total costs, the focus will rely on the differences between powders and concentrates, and only those processes being different will be taken into consideration. The studied TQ will be a relative measure, that is, all scenarios will be compared to the same reference and will include production, transport, required cooling for transport , storage, required cooling for storage and waste.

Regarding the choice of the average quality as a KPI, it is a crucial matter for concentrates to provide similar quality characteristics than powders under similar storage conditions to fairly compare the TC. Therefore, the arithmetical mean of all delivered batches will be calculated by the simulation program.

Furthermore, two different model parameters will be included in the simulation study: delivery frequency from the farm to the dairy plant, as well as transportation and storage cooling; both of them having to possible configurations.

The delivery frequency from the farm to the dairy represent the most common practice in the sector (Bylun, 1995), and can be either one or two days (1d or 2d). This will affect the milk quality when arriving at the dairy, that is, milk delivered to the dairy daily will have better quality but will require more transportation efforts; on the other hand, farms delivering milk only every two days will provide lower quality but transportation savings.

In addition, the transportation and storage cooling option or the room temperature option (room temperature: RT or cooling: COLD) will have similar consequences regarding the tradeoff between quality and energy or transportation efforts.

In brief the scenario and parameters overview is shown in table 4.1, where scenarios can be identified by their code according to the above mentioned abbreviations as follows: process.deliveryFrequency.cooling

Table 4.1 Scenario and parameters overview

	S0	S1	S2	S3	S4
1 day		S1. 1d. RT S1. 1d. COLD	S2. 1d. RT S2. 1d. COLD	S3. 1d. RT S3. 1d. COLD	S4. 1d. RT S4. 1d. COLD
2 day	S0. 2d. RT	S1. 2d. RT S1. 2d. COLD	S2. 2d. RT S2. 2d. COLD	S3. 2d. RT S3. 2d. COLD	S4. 2d. RT S4. 2d. COLD

4.2.3 Overall model assumptions

The model has been implemented under some general assumptions, which apply to the total simulation time and are common for all the processes.

Firstly, the time format used in the implementation for all processes is given by Plant Simulation as follows:

DDDD:HH:MM:SS (in words, days:hours:minutes:seconds).

Furthermore, the simulation run starts with an empty system, that is, the first part enters the system at time 0:00:00:00, what implies that all the machines, buffers, trucks, etc. are empty, included the DC. Since that means there is no safety stock hold at the DC and it would be possible for the first orders to have stock outs, the data collection from the simulation run should not be taken into account until the system has reached a steady state. This happens after the 25th day in the simulation time.

The simulation horizon should be 1 year (365 days) in order to compare annual costs, nevertheless, in order to avoid the consideration of the data collected during the unstable period, the time horizon is extended.

Hence, the acceptable error is set at 2%, meaning the simulation should run for at least 1250 so that the unsteady period represents 2% of the simulation time. Finally, simulation time horizon is fixed at 4 years (1640 days) with 1.71% error; simulation results are then divided by 4 to obtain the mean annual data.

Regarding the fact, that no real data is available yet, some of the processes and their characteristics are implemented by using stochastic models.

This applies especially to three kinds of processes:

- Processes that are not computer controlled for instance biological processes such as milking; in terms of milk quantity obtained at the farm and initial temperature of the milk.
- Production process variability is considered by using stochastic models for the achieved temperature of a batch when being processed or for the actual % of volume reduction.
- Customer demand, regarding order quantity.

Since probability distributions are computer-generated, the stochastic models used in the simulation are based on pseudo-random numbers. These are created by seed values, which generate independent random number streams by using different seed values.

Because there is no available data yet, this model uses only normal distributions that may represent reality, what should be later changed to fit a specific SC. Normal distributions are implemented in Plant Simulation as follows: [Stream, Mean, Std. Deviation, Lower Bound, Upper Bound]

Furthermore, in order to properly describe the dynamic behavior of the SC, some auxiliary objects have been implemented, meaning they do not represent any real process, but are necessary for the adequate simulation calculations and functionality. These objects are included in the methods and have a processing time of 0,1 seconds so that they can fulfill their auxiliary mission but do not compromise the timing of the model.

Moreover, three specific parameters are assumed to be constant for the model: room temperature is set at 20ºC, what also applies for transportation and storage facilities without cooling installations; machines are available 99,5% of the simulation time in order to take into consideration possible break downs, reparation and maintenance tasks, etc.; and finally, the model is based on the assumption that trucks travel at a mean speed of 80 km/h; that is transportation time can be defined by the length of the transportation process, or in other words, by the distance between facilities.

4.2.4 Quality models

As explained in section 2.2, the Arrhenius model will be used for the quality decay during the simulation run.

As determined by Fu *et al.* (1991), the microorganism *Pseudonomas fragi* is a good indicator of the quality level because of its prevalence and active growth in dairy products.

In the research they determined an empirical model for milk flasks incubated at 4ºC during 50 hours and then cooled in four different scenarios. The obtained kinetic parameters provided following model with a 95% confidence interval and R^2=0,984:

$$\Delta \mathrm{q} = \exp[-\exp(30{,}10) \cdot \mathrm{t} \cdot \exp(-8{,}9 \cdot 10^3/\mathrm{T})]$$

As for the powders and concentrates, and because of the lack of empirical models, the activation energy was altered.

The activation energy measures the exponential temperature dependence, in other words, how the substance reacts to temperature (Zanoni and Zavanella, 2011).

Because this reaction is highly dependent on the substance’s water content (Bylund, 1995), it is reasonable to use a lower E_a value for powders than for concentrates. Without any scientific basis and only for this particular simulation study, so that the model consistency could be proved and a reasonable outcome would be obtained (otherwise all the batches could finish the process with zero quality level), following E_a values were used:

E_a(powders)=23,1 kJ/mol: $\Delta q = \exp[-\exp(23{,}10) \cdot t \cdot \exp(-8{,}9 \cdot 10^3/T)]$

E_a(powders)=28,1 kJ/mol: $\Delta q = \exp[-\exp(28{,}10) \cdot t \cdot \exp(-8{,}9 \cdot 10^3/T)]$

The value for powders (difference to the milk value of 7 kJ/mol) was fixed arbitrarily to represent a 98% dry matter content , so that for a 30% dry matter content a difference regarding the milk value of approximately 2 kJ/mol was calculated, that is 28,1 kJ/mol.

4.3 Simulation model

4.3.1 Material flow objects

The different processes included in the model are represented by several material flow objects, which simulate the flow of materials through the chain. A brief description of the most important ones is given according to the Tecnomatix PlantSimulation Help:

- Source

The Source produces all kinds of *MUs* in a single station, has a capacity of one and no processing time.

❖ Drain

It is the object responsible for removing the parts (produced by the source) from the model after they have been processed.

❖ SingleProc

Production processes and machines are represented by a SingleProc, a single station for processing one part, which it received from its predecessor, then processed and finally passed on to the successor.

In order to properly represent the characteristics of each one of the production processes, following attributes are defined for all the SingleProcs (see table 4.2):

Table 4.2 Attributes of the SingleProc

Attribute	**Description**
Processing time	Required time to process one part
Failure rate	All machines are available 99,5% of the total time
Temperature	Working temperature of the machine in ºC
Weight reduction	Weight reduction in % of the initial volume when leaving the machine

❖ Buffer

The storage elements are represented by Buffers, accomplishing tow different missions: temporarily holding parts when the following component fails and passing parts on when the preceding component stops working. Its attributes are explained in table 4.3 and have several similarities with the SingleProc's attributes.

Table 4.3 Attributes of the Buffer

Attribute	**Description**
Processing time	Required time to process one part (if necessary)
Temperature	Working temperature of the machine in ºC
Exit strategy	All Buffers use *Queue* behavior (FIFO strategy)

❖ Track

Transportation distances between parties are represented by Tracks, which have a defined direction (one way) and following attributes (see table 4.4):

Table 4.4 Attributes of the Track

Attribute	**Description**
Length	Transportation distance in meters
Speed	Speed of the trucks placed on the track in m/s
Temperature	Transport temperature in ºC

❖ Connector

Connections between two objects in the same frame on which the parts move from object to object; as well as connections between an object and an exit or entrance of a frame are represented by Connectors. These also show the direction of the connection.

❖ Interface

Transitions between frames are modeled with the object Interface, which are the places at which the *MUs* pass from one frame to another in the simulation model.

Each material flow object has its own identifying icon, as shown in figure 4.4:

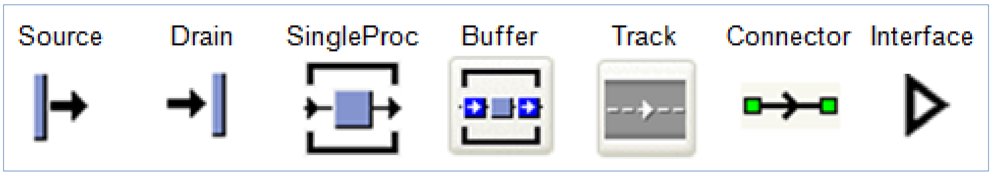

Figure 4.4 Flow object icons overview

4.3.2 Moving units: entities and containers

In order to represent the different stages and the dynamics of the model, two kinds of moving units (MUs) are used: entities and containers.

- Entities

The product flow is represented by entities. These are moving material flow objects without loading capacity that move through a plant on the material flow objects proper, representing parts being produced, processed and transported (Tecnomatix Plant Simulation Help).

In this case, entities will represent the liquid milk, the milk powders, the milk concentrates or any intermediate state involved in the process.

For that purpose, entities are defined with following embedded attributes (see table 4.5).

In addition, other attributes are related to entities with the only purpose to store data for concrete methods. These do not represent any real attribute and will be described in the corresponding method explanation along section 4.2.

Table 4.5 Attributes of the entity

Attribute	Description
Temperature	Current Temperature of the entity in ºC
Weight	Weight of the entity in kg
Trace	Traceability table containing the complete process data
Quality	Relative quality level regarding the initial quality in %

Because of the large number of entities being processed at some point of the system at the same time, it is not possible to address the entities directly; hence, they will be called by the anonymous identifier “@” in entrance or exit controls.

- Containers

Now turning to the containers, these are moving material flow objects for transporting other MUs (entities in this case), which can be used to model pallets, bins, boxes, etc. or as in this study, trucks.

During a simulation run Plant Simulation passes the container along from material flow object to material flow object along the existing connectors or according to programmed methods.

In line with the entities attributes, the containers’ attributes are described in table 4.6.

Likewise the entities, other attributes for internal methods’ calculation are implemented and the addressing of the containers is made by the anonymous identifier “@”.

Table 4.6 Attributes of the container

Attribute	Description
Capacity	Truck loading capacity (number of entities)
Customer	Destination of the product (only for trucks between DC and Customer) *(*)*

(*) Two different definitions of the container are implemented in the class library: the container provides transportation between the farms and the dairy as well as from the dairy to the DC, and the container_c, represents the transport from DC to the corresponding final customer. Characteristics like transportation temperature and transportation time are included in the road object; loading and unloading times in the process itself and will be explained in the corresponding section.

4.3.4 Implementation of quality models

The quality of the milk products should be intrinsic the process as introduced in the literature review section, and for that purpose, the new quality level is calculated by the program every time an entity leaves a material flow element, that is a SingleProc or a Buffer.

Thus, a global method is implemented in the class library and linked to all the exit controls where needed. The method is named “TQSL” (Temperature, Quality, Shelf-Life) and is structured in following steps:

1) Declaration of local variables (see figure 4.5):

```
is
    dT : real;
    dQ : real;
    t : time;
    i: integer;
    days: real;
do
```

Figure 4.5 TQSL: local variables

In order to program the necessary calculations some auxiliary variables are implemented only for this method, for instance:

- dT: differential of temperature (T) to include process variability
- dQ: differential of quality between the quality of the processed entity and the quality before entering the machine.
- t: time interval in which the entity has been processed in the machine.
- i: local counter for the traceability table.
- days: conversion variable for t (seconds to days).

2) Calculation of time and temperature

As explained in section 4.2.4, the chosen quality model calculates the quality variation in a time interval. In order to calculate this interval, every material flow object with the exit control TQSL has also an entrance control “entrance” (see in figure 4.6) in which the point in time when an entity (@) triggers the control is saved in the entity’s attribute “in”. This attribute was not included in the previous section since it is only used for this particular calculation.

```
is
do
    @.in:=.powder.CHAIN.EventController.simTime;
end;
```

Figure 4.6 Method entrance

Afterwards, when the exit control is triggered and once the local variables have been declared, the time interval is calculated and stored in variable “t” [$t = t_{out} - t_{in}$]. Then, it is transformed from seconds to days and stored in variable “days”, as shown in figure 4.7.

```
do
    -- time
    t:=.powder.CHAIN.EventController.simTime-@.in;
    days:=t/86400;

    -- Get T from machine + random variation
    dT:=z_normal(1,1,0.3);
    @.Temperature:=?.Temperature-dT;
```

Figure 4.7 TQSL: time and temperature

The next step consists on getting the temperature information from the processing machine or buffer (?) and store it in the entity’s (@) temperature attribute with some random variation provided by a normal distribution calculated in dT ~ N (1;03) and seed value 1 (see figure 4.7).

3) Calculation of the new quality level

Thereafter the quality change (dQ) can be calculated according to the quality model formula, as shown in figure 4.8; and multiplied by the previous quality level of the entity (stored in the attribute “Quality”: @**.**Quality); in order to obtain the new quality level.

```
-- Calculate new Quality
dQ:=exp(-exp(30.1)*days*exp(-73900/(8.31*(@.Temperature+273))))
@.Quality:= @.Quality *dQ;
```

Figure 4.8 TQSL: quality level

4) Calculation of the weight reduction

It is also in this method where the weight reduction, as a consequence of the water extraction in the evaporation and drying processes, is taken into account.

```
-- Weight reduction + random variation
@.w := @.w*(1-?.Wred*z_normal(4,1.01,0.001));
```

Figure 4.9 TQSL: weight reduction

The new weight is calculated and stored by multiplying the current weight (@**.**w) by the reduction factor stored in the machine (?.Wred) and then multiplied by a normal distribution in order to add process variability, with seed value 4, mean 1,01 and standard deviation 0,001 (see figure 4.9).

5) Traceability table

Finally, all the process information and data is recorded in the traceability table owned by each entity (see figure 4.10). First of all, the last row containing information is located by means of the dimension of the current table (@.Trace.yDim) and then the last process data is recorded in the next row (@.Trace.yDim+1), by columns:

- processing machine (?.name)
- process starting time (@.in)
- process finishing time (simTime)
- process T (?.Temperature)
- final quality level (@.Quality)
- final weight (@.w)

```
    -- Update traceability table:
    -- 1)find first blank row; 2)write new tracebility data
    i:=@.Trace.yDim+1;
    @.Trace[1,i]:=?.name;
    @.Trace[2,i]:=time_to_str(@.in);
    @.Trace[3,i]:=time_to_str(.EventController.simTime);
    @.Trace[4,i]:=?.Temperature;
    @.Trace[5,i]:=@.Quality;
    @.Trace[6,i]:=@.w;

    -- Move to next station
    @.move;
end;
```

Figure 4.10 TQSL: traceability table

Finally, all exit controls need to include the command @.move in order for the entity or moving unit being processed to move to the next station; otherwise the object is hold by the exit control and not allowed to continue with the process.

4.3.4 Supply Chain (top frame)

As introduced in chapter 3, the first frame found in the class library, that is the top layer, is named *CHAIN* and represents a general SC for milk powder production in Germany.

The frame CHAIN has no function itself other than containing all the other process-specific frames, including (see Figure 3.1):

- Farms (five in total, 3 regular farms and 2 big farms)
- Transportation from each farm to the dairy plant
- Dairy plant
- Transportation from the dairy plant to the DC
- Distribution center
- Transportation from the DC to the appropriate customer
- Customers (two different customers with different requirements)

In addition, the frame CHAIN contains several other elements that keep the cohesion of the simulation calculations.

For instance, the Event Controller (EC) is placed as well in the top layer and it coordinates and synchronizes the different events taking place during a simulation run within all frames (Tecnomatix Plant Simulation Help). It is here where the simulation horizon, the simulation speed and the reset characteristics can be set. Moreover, the init Method is placed as well in the frame CHAIN, since it is a general Method activated by the EC; which, in this case, initializes the global model variables (see 4.3.11).

4.3.5 Farms

The simulation model contains five farms, three of them considered regular farms, regarding the milk quantity they provide and two of them implemented in the class library as BigFarms, because of a grater milk quantity production.

The structure of these frames consists of the source (*cows*), which generates the entities representing the milk bulks, the chilling facilities where the milk is treated and storage after the milking of the cows, and the method *attributes* (see Figure 4.11).

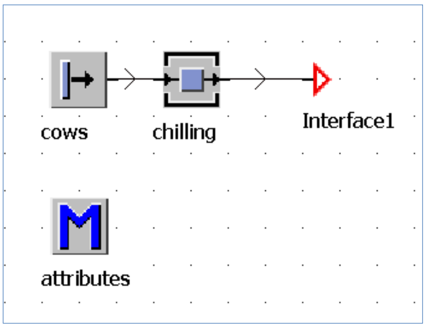

Figure 4.11 Frame Farms

For a better understanding of the simulated process, each stage is described in detail:

- ❖ Cows (source)

For this study, the starting point of an entity in the system is considered to be the moment right after the cow milking is finished, that is the first moment in time when the milk is available.

Thus, the source represents this milking output; it produces *MUs* in a single station but has a capacity of one and no processing time (Tecnomatix Plant Simulation Help). In this case, the source produces only one kind of entities (named milk in the class library).

According to the Dairy Handbook by Bylund (1995), usual practice is to milk cows by a machine once a day, fact that is represented by the source interval, which is set to one day (1:00:00:00) equally for all the farms (see Figure 4.12).

To avoid the blocking of the reception at the dairy and add some realism into the model, each source starts producing milk entities at a different time, which is specified in the start box of the source dialog (see Figure 4.10). This represents the fact that due to different milking systems, machines, number of cows in the farm, time of the milking, etc. every farm will finish the milking process at a different time.

Hence the farms start at time 0, 3 hours and 8 hours for the regular farms, as well as 6 and 10 hours for the bigger farms.

After the entity "milk" is produced it leaves the source triggering the exit control driven by the method attributes.

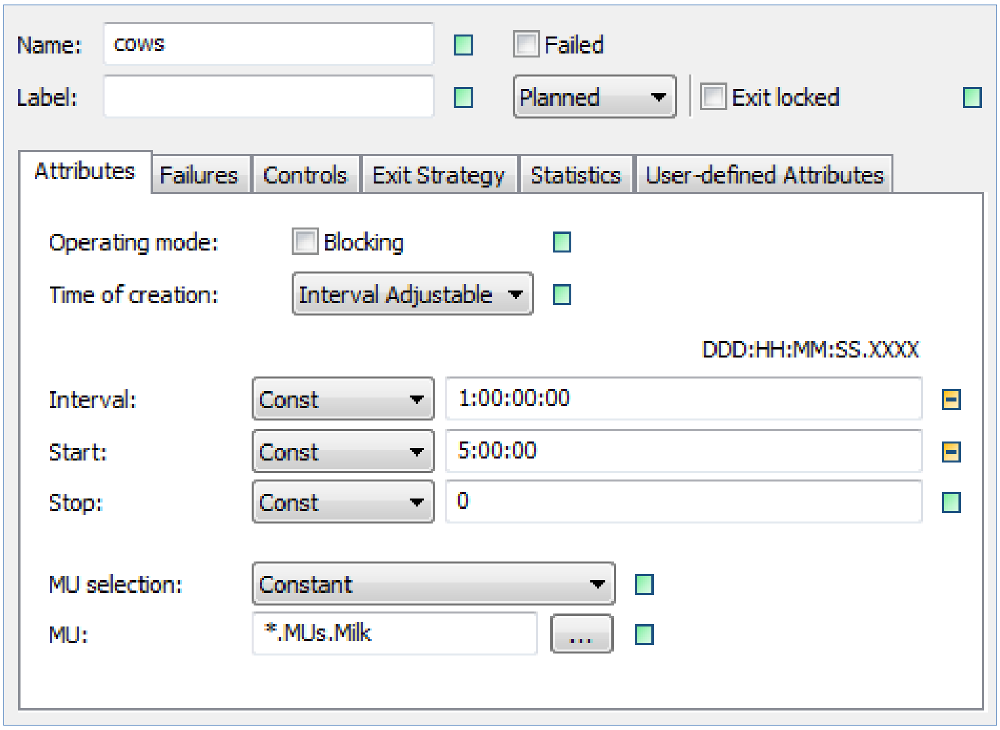

Figure 4.12 Source (cows) dialog example

❖ Attributes (method)

Once the entity has been introduced in the system, some of its attributes need to be set, as in the case of temperature and weight. These are implemented in the method "attributes" shown in figure 4.13.

As for the temperature, it is set to 37ºC, which is the temperature at what milk leaves the udder (Bylund, 1995). Nevertheless, cows corporal temperature is approximately 38,6ºC which in addition to different milking processes and different cow characteristics can lead to some temperature variation. In order to take this fact into account, some random temperature variation is added to the 37ºC, following a normal distribution with seed value 3, mean 0,1ºC and standard deviation 0,05ºC.

Turning to the weight attribute, and under the assumption that a regular farm has 500 cows producing between 28 and 30 liters each, every entity leaving the milking process represents the outcome of the whole farm and is approximately 1500 liters.

Thus, raw milk density is 1,028 kg/l at a temperature of 38ºC (Bylund,1995) that is 1466,3 kg. All in all, the attribute weight is implemented by the addition of the initial estimation of 1500 kg to a normal distribution with seed value 2, mean 100 kg and standard deviation 35 kg.

Because the name “weight” is used by Plant Simulation for the units’ definition and not allowed to be used anywhere else, in the programming of the methods from now on, the attribute weight is called by the name “w”.

```
is
do
    @.Temperature:= 37 + z_normal(3, 0.1, 0.05);
    @.w:= 1500 + z_normal(2, 100, 35);
    @.move;
end;
```

Figure 4.13 Method attributes

- Chilling (SingleProc)

As explained by Bylund (1995), the milk should be chilled to below 4°C immediately after milking and be kept at this temperature all the way to the dairy; that is the cold chain cannot be broken. Otherwise micro-organisms in milk would start to multiply affecting milk quality (see figure 4.14).

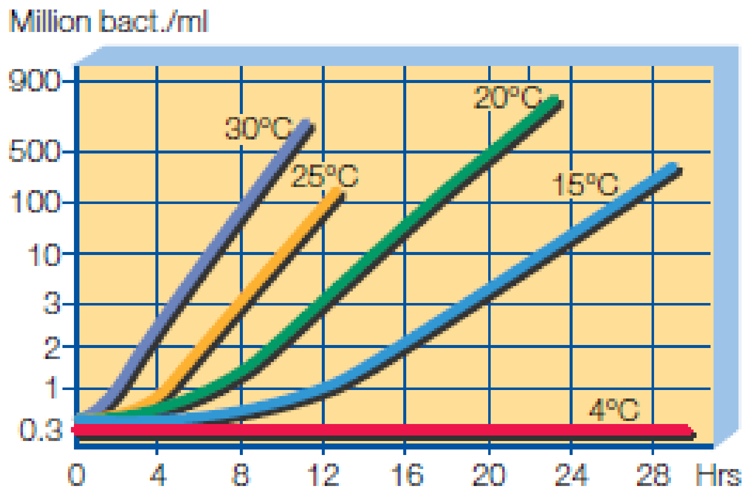

Figure 4.14 Influence of temperature on bacterial growth in raw milk (Bylund, 1995)

This is represented in the simulation by the chilling process, which is a SingleProc machine, with a processing time of 1 hour (1:00:00), entrance control “entrance” and exit control “TQSL”.

The chilling process represents the need for a fast cooling below 4ºC, which is more energy intensive; the rest of the storage time (until the milk is picked up for transportation) is also at the same temperature but consists only in maintaining it and is included in the transport frame.

Once the entity leaves the chilling station and completes the exit control TQSL it is passed on by the corresponding interface to the next frame, in this case, the fist transport frame.

4.3.6 Transport from farm to dairy

The transport frame consists of two sub-frames, the loading and the unloading process, in addition to a track object in between them. Because the simulation model includes five farms under the assumption that they have all different locations, this three parts structure is repeated for each farm, representing the different transport conditions for each one of them (see figure 4.15).

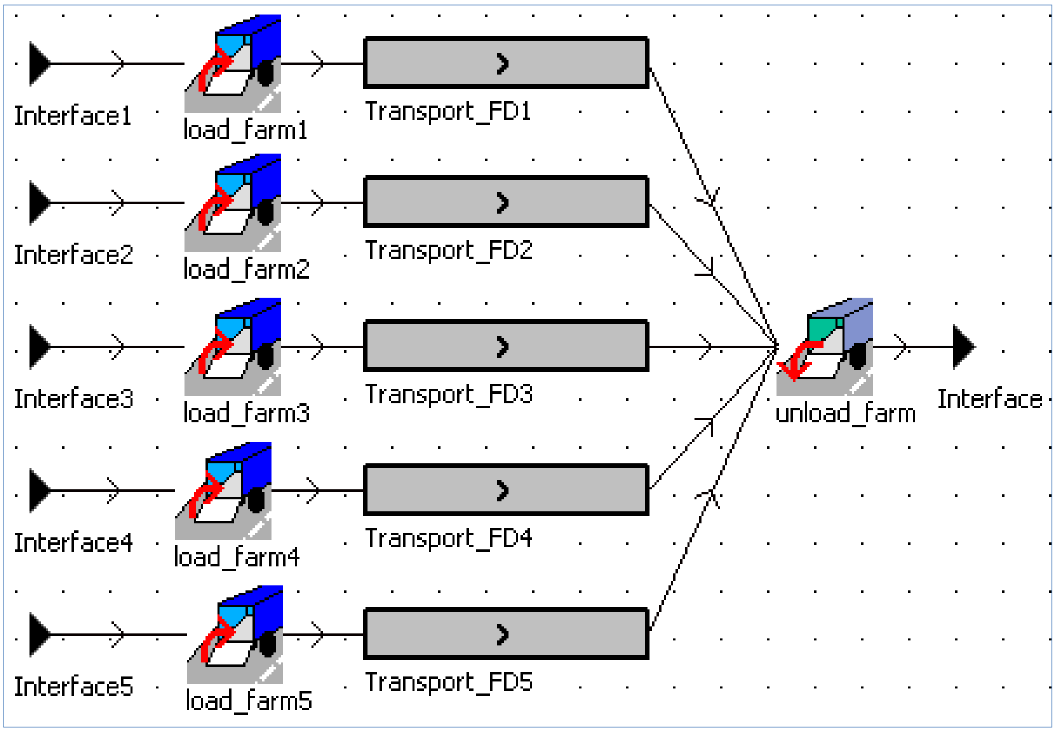

Figure 4.15 Frame Transport (farm to dairy)

As mentioned above, the transport frame contains three parts, which represent three different processes happening in three different places, thus, the sub-frame load_farm models the loading dock at the farm, the track models the road transportation and finally, the sub-frame unload_farm represents the unloading dock at the dairy plant.

- ❖ Loading process (load_farm)

The loading process is modeled by a Buffer and three SingleProcs, as shown in figure 4.16.

The Buffer represents the storage time after the chilling process and the milk pick up for transportation, hence the temperature in the Buffer is kept at 4ºC as explained in the chilling process.

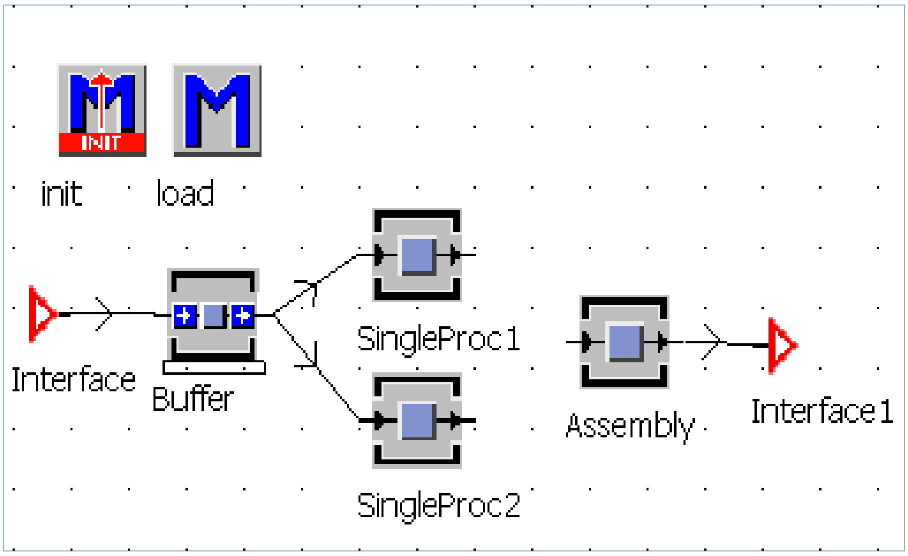

Figure 4.16 Frame load_farm

When leaving the Buffer, entities are delivered to SingleProc1 if the delivery frequency parameter is one day, or alternatively, to SingleProc1 and SingleProc2 if the delivery frequency parameter is two days.

As explained in the general model assumptions, these two objects do not represent any real process and have only an auxiliary function with 0,1 seconds processing time, thus, they do not have any entrance or exit controls and do not affect the quality of the entities.

Afterwards, when the milk bulks have to be transported, the milk entities in SingleProc1 and/or Singleproc2 are loaded onto a container placed in the SingleProc named “Assembly”. This station has a processing time of 20 minutes and represents the actual loading process.

Because the entities receive the move-order from the method load, it is not necessary to bind the SingleProcs and the Assambly with a connector.

Finally, the truck, modeled by the container and the entities, is passed by the interface to the next frame.

In order to control the loading process, two continuously running methods are used: firstly the init method, which function is to execute the method “load” as soon as the simulation run begins (see figure 4.17).

```
is
do
    ref(load).methcall(0);
end;
```

Figure 4.17 Method init (load_farm)

The second and more important method is the method “load”, shown in figure 4.18. The first condition in the method holds it until the entities are ready to be transported (once a day or every two days depending on the scenario) and until the previous truck has left (only as a safety command to assure the functionality of the model in extreme cases). Once all the conditions are fulfilled, the arrival of a truck to the loading dock is modeled by creating a container at the "Assembly".

For the purpose of keeping track of the number of trucks needed in each scenario, a global variable is used and updated here, what will be described in detail in section 4.3.11.

Finally, the milk entities are loaded onto the container by moving them on the Assembly's content (container) and will leave the frame by the Interface.

The last command of the method starts the method execution again, that is the method starts running from the top line again; in other words, the method will then wait until the next shipment is available and then start the process again.

```
is
do
    -- wait until parts available & assembly station free
        Waituntil SingleProc1.occupied
        and SingleProc2.occupied
        and Assembly.empty prio 3;

    -- create a new main part on the assembly station
        .MUs.Container.create(Assembly);
        .powder.trucks:= .powder.trucks + 1;

    -- assemble all parts to the main part
    SingleProc1.cont.move(Assembly.cont);
    SingleProc2.cont.move(Assembly.cont);

    -- starts method execution again
    self.methcall(0);
end;
```

Figure 4.18 Method load

❖ Road transport (Transport_FD)

The transportation between Farms and Dairy is modeled by a track object and is characterized by two parameters: transportation time and transportation temperature.

As for the transportation time, as already explained in the general assumptions, the mean speed of the trucks is set at 80 km/h, so that the actual distance between the farm and the dairy plant is what determines the transportation time. The distances and transportation times are shown in table 4.7:

Table 4.7 Transportation times and distances

	Time [hh:mm:ss]	Time [h]	Distance [km]
Farm1	3:20:00	3,33	266,67
Farm2	2:05:00	2,08	166,67
Farm3	2:46:40	2,78	222,22
BigFarm1	1:40:00	1,67	133,33
BigFarm2	2:36:15	2,60	208,33

Regarding the transportation temperature, it is set at 4ºC for all the farms, as for the chilling and storage processes.

Similarly to the general method entrance, tracks a the same entrance method, called “entrance_transport” with the only difference that the information is stored at the containers attribute “in” and not in the entities.

As for the exit control, the necessary information is also stored in the containers attributes (see figure 4.19), for instance the transportation time is stored at the container’s attribute (@) “transport” and the transportation temperature is obtained from the track's attribute (?) "Temperature" and is then stored at the container's attribute (@) "Temperature". The container's attributes will be used in the next frame to update the quality level of the entity by using the method TQSL.

```
is
do
	-- time
	@.transport:=.powder.CHAIN.EventController.simTime-@.in;

	-- Get T from road
	@.Temperature:=?.Temperature;
	@.move;

end;
```

Figure 4.19 Method Tt_Transport

As usual in exit controls, the final order is to move, in this case the container, to the next station, which is here the "unload" frame.

- ❖ Unloading process (unload_farm)

The last step in the transportation frame is the unloading process, which is implemented in the frame named "unload_farm". It consist of a SingleProc, a Buffer, a Drain and the unload method, as shown in figure 4.20.

Once the loaded container enters the frame by the interface, it is placed in the SingleProc, representing the dock in order to unload the milk. After the processing time of 30 minutes is over, the exit control "unload" is triggered (see figure 4.21).

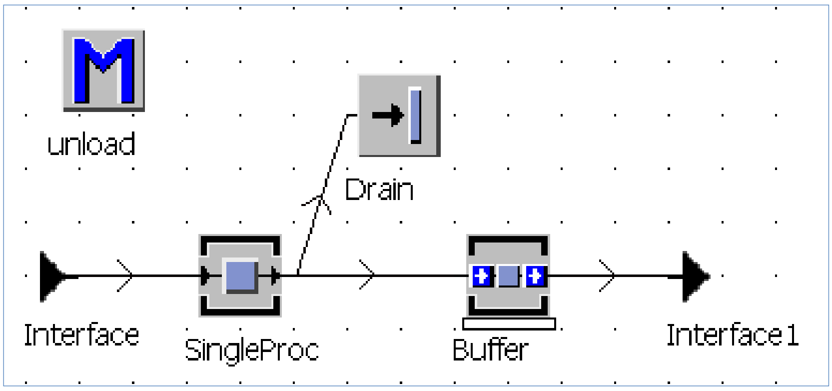

Figure 4.20 Frame unload_farm

The method “unload” consist of a while-loop, with the condition that at least one moving unit is placed at the SingleProc. When this condition is true, for instance when a loaded container arrives to the dairy dock, the method gets the transportation T from the container (@.Temperature) and stores it in the entity’s attribute (@.cont.) “T_transport. The same procedure is used for the transportation time: the container’s information (@.transport) is stored in the entity’s attribute “transportTime”.

Finally, the entities (@.cont) are moved to the Buffer so that they can continue the process and the container (@) is moved to the drain.

```
is
do
    while @.numMU > 0 loop
        @.cont.T_transport:=@.Temperature;
        @.cont.transportTime:= @.transport;
        @.cont.move(Buffer);
    end;
    @.move(Drain);
end;
```

Figure 4.21 Method unload

Once the entities are in the Buffer, which is an auxiliary object with no processing time, the exit control “Qtransport” is triggered, which is very similar to the method TQSL, but instead of using the characteristics of the current machine (here the auxiliary Buffer) it uses the information stored in the entity to update both, quality level and traceability table to finally get to the dairy plant by the interface.

4.3.7 Dairy plant

The dairy plant frame is divided in three parts as shown in figures 4.22 and 4.23: first the milk arriving from the dairy is tested for appropriate quality conditions, which is implemented in the sub-frame “QControl1”. The second part of figure 4.19 is the production process itself, where the processing for powders and concentrates is different and therefore implemented in two models. Finally, in figure 4.20 an auxiliary process is implemented, which does not represent any real process but it is used to model the transition from the milk entities to the 25kg-bag entity.

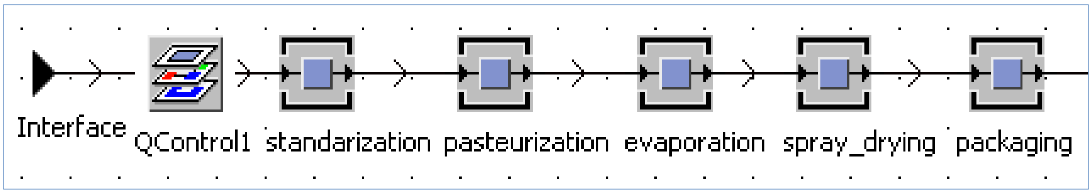

Figure 4.22 Frame Dairy (I)

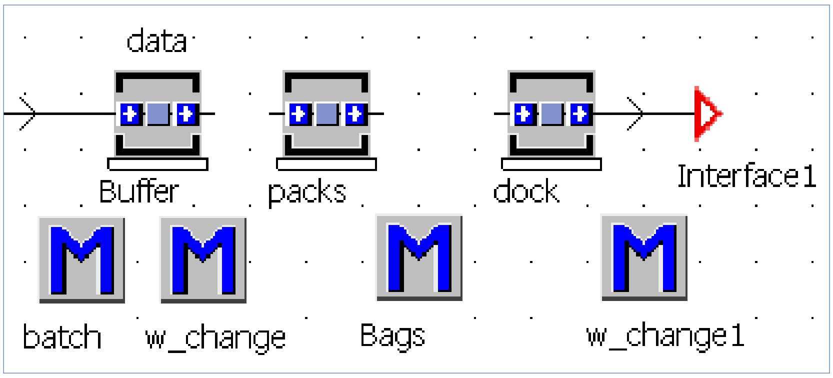

Figure 4.23 Frame Dairy (II)

- Quality Control

Regarding the quality control sub-frame, it is composed of a buffer, a SingleProc, a drain and two methods (see figure 4.24).

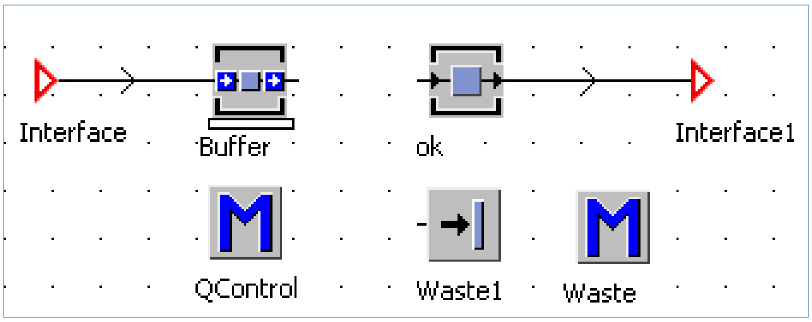

Figure 4.24 Frame QControl

The entities coming from the transportation frame are then moved to the auxiliary Buffer (capacity 1 and processing time 0,1 seconds). The exit control QControl is then triggered and the entity is passed on either to the SingleProc “ok” or to the drain “Waste1” according to its quality level (see figure 4.25). Only batches with a quality level equal or greater than 95 can continue the process; entities not reaching this level are considered waste, since it is important to have high quality raw milk before starting the production process (Kessler, 2002).

```
is
do
    if @.Quality >= 95.0 then
        @.move(ok);
    else @.move(waste1);
    end;
end;
```

Figure 4.25 Method QControl

Entities with good quality level leave the quality control through the interface in order to continue the process, batches removed from the process are taken into account by the drain's entrance control "waste", where the global variable "kgWaste" is updated (explained in detail in section 4.3.10).

- Dairy production process

Each one of the processing steps is modeled by a SingleProc, which its attributes processing time and temperature and respectively the entrance and exit controls "entrance" and "TQSL".

The above mentioned attributes are implemented differently for each scenario regarding process steps, processing times and processing temperatures, as shown in table 4.8 (Kessler, 2002).

Nevertheless, the four different processes are still under research and no empiric data exists yet. For that reason, the times and temperatures used are only general descriptions on the different treatments by Kessler (2002) and approximate data.

In addition, the first and final step are equal for all scenarios, that is the standardization, with a processing time of 21 minutes at a temperature of 55ºC; and the packaging, with a processing time of 2 minutes at 28ºC.

Table 4.8 Processing characteristics

		time [s]	temperature [ºC]
S0	Pasteurization	1200	85
	Evaporation	720	70
	spray drying	180	195
S1	pasteurization Uht	15	135
	aseptical concentration	900	45
S2	MF filtration	120	55
	KZE Heating	90	95
	aseptical concentration	900	45
S3	Concentration	720	49
	MF filtration	120	55
	KZE Heating	90	95
S4	MF filtration	120	55
	Concentration	720	42
	KZE Heating	90	95

❖ Auxiliary packaging process

Once the entities have been processed and packaged, the entities' attributes have to be changed in order to reflect the current product characteristics, as shown in figure 4.21.

Firstly, after the packaging process, the entities are moved to the auxiliary Buffer triggering the entrance control "batch" (see figure 4.26), which divides to powder or concentrate batch in 10 kg groups. For this purpose, the result of the entire division of the entity's weight (@.w) by 10 (k) is the amount of new entities that will be created in the Buffer (implemented in the for-loop), the rest of the entire division, as well as the decimal part of the weight attribute are considered waste produced along the production process.

```
is
    n: integer;
    k: integer;
do
    -- 10 kg grouping, rest considered waste
    n:=abs(@.w)-1;
    k:=n//10;
    .powder.kgWaste:=.powder.kgWaste+(@.w-abs(@.w)-1+n\\10)

    -- safe traceability table in aux var data
    .powder.data:=@.trace;

    for local i := 1 to k loop
        .MUs.Milk.create(Buffer);
    next;
    @.vernichteBEs;

end;
```

Figure 4.26 Method batch

As an illustration, if an entity with 478,35 kg arrives to the auxiliary buffer, when triggering the batch method, these will be the values in the method:

n = 479 – 1 = 478 kg

k = 478 // 10 = 47

kgwaste* = 478,35 – (479 – 1) + (478 \\ 10) = 0,35 + 8 = 8,35 kg

This means, that a maximum of 9,99 kg could be considered waste in batches with mean weight of 251,6 kg. In the worst case scenario, 3,97% of waste is generated in the production process.

This waste percentage can be narrowed if necessary, by smaller grouping; that is dividing in groups of 5 kg or even smaller. In this study it was not possible to do so because of the educational license restriction to 1000 objects in the model.

Once the new k entities are created in the Buffer, the former milk entity is destroyed and eliminated from the model. Before that, the traceability table is stored in the variable “data”. By leaving the Buffer, the exit control “w_change” is triggered where the new entities are given the necessary attributes: the weight (@.w) is set at 10 kg and the traceability information (@.trace) is copied from the table stored in “data” (see figure 4.27).

```
is
    i:integer;
do
    @.w:=10;
    @.trace:= .powder.data;
    i:=@.Trace.yDim;
    @.Temperature:= @.trace[4,i];
    @.Quality:=@.trace[5,i];
    @.move(packs);

end;
```

Figure 4.27 Method w_change

To continue the process, the entity also needs a quality level and its temperature, information that can be copied from the last stage representing a real part of the process, here the packaging. This data can be obtained from the traceability table by using a counter (i) to access the last row of the table (@.Trace.yDim) and then coping the data stored in the temperature and quality columns (4 and 5) to the temperature (@.Temperature) and quality (@.Quality) attributes.

Finally, the new complete entities are moved to the buffer “packs”, where they are grouped into 250 kg batches, which represent a pallet containing 10 bags of 25 kg each. For that purpose, the exit control “Bags”, shown in figure 4.28, holds the 10 kg entities in “packs” until the buffer capacity of 25 is reached, that is a total weight of 250 kg is grouped together, and then creates a new entity in the buffer “dock”.

By using a similar procedure to the one in method w_change, traceability table, current temperature and quality level are stored in the new entity’s attributes. Finally, the 25 (packs.capacity) old 10 kg entities are deleted.

```
is
    j: integer;
do
    waituntil packs.full prio 3;
    .MUs.Milk.create(dock);
    j:=@.trace.yDim;
    dock.cont.trace:=@.trace;
    dock.cont.Temperature:=@.trace[4,j];
    dock.cont.Quality:=@.trace[5,j];

    for local i := 1 to packs.capacity loop
        packs.cont.delete;
    next;
end;
```

Figure 4.28 Method Bags

Once the 250 kg entity leaves the buffer “dock”, the exit control “w_change1” is triggered and the entity’s weight (@.w) is set at 250 kg (see figure 4.29).

```
is
do
    @.w:=250;
    @.move;
end;
```

Figure 4.29 Method w_change1

Finally, the auxiliary process is finished and the entities are moved to the next station on the next frame.

A schematic diagram of the auxiliary process is shown in figure 4.30:

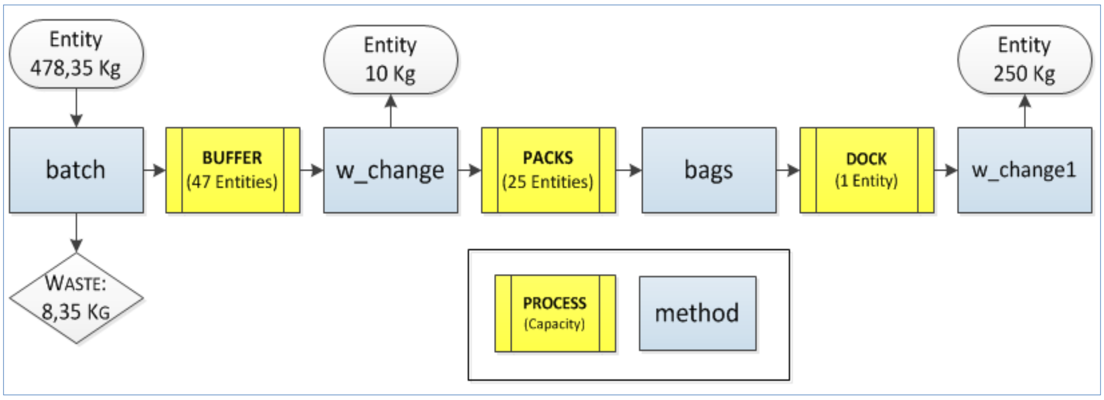

Figure 4.30 Packaging auxiliary process

4.3.8 Transport from dairy to DC

The frame “Transport” is used again to model the transportation between dairy and DC, but unlike the frame in section 4.3.6, only one load and one track are needed in addition to the unload process (see figure 4.31).

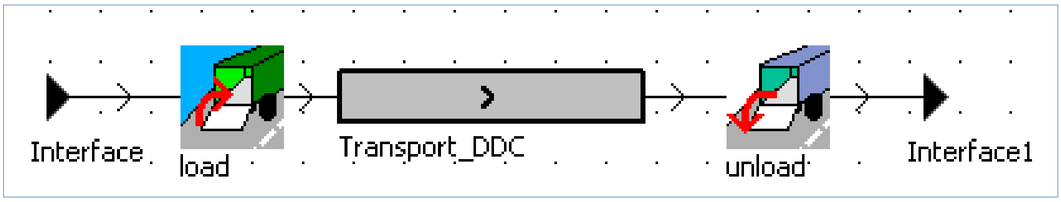

Figure 4.31 Frame Transport1

❖ Loading process (load)

The loading process at the dairy is modeled in the frame “load” and consists of two buffers, two SingleProcs, one source and three methods, as shown in figure 4.32:

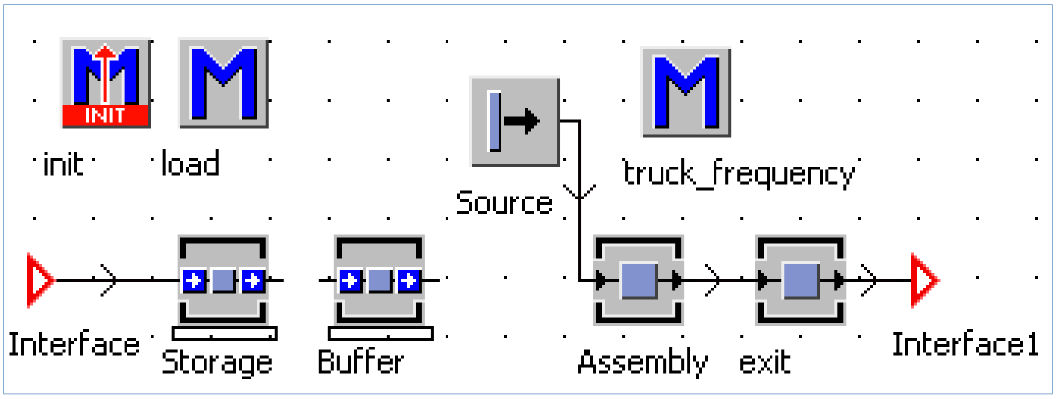

Figure 4.32 Frame load

The 250 kg entities arrive to the Storage by the interface and are stored until the method load moves them to the Buffer, which represents the order set up in pallets.

The method “load” (see figure 4.33) is a continuously running method initiated by the init method that holds entities in the “Storage” buffer until a container is available at the “Assembly” (Assembly.occupied), until the Buffer is empty (Buffer.empty) and until there are enough 250 kg entities in the “Storage” to fill the container’s capacity (Assembly.cont.capacity).

```
is
do
    -- wait until parts and truck are available
    Waituntil Assembly.occupied
    and Buffer.empty
    and Storage.numMU>=Assembly.cont.capacity prio 3;

    Buffer.capacity:=Assembly.cont.capacity;

    -- batch set up + load onto container
    for local i := 1 to Assembly.cont.capacity loop
        Storage.cont.move(Buffer);
    next;

    for local i := 1 to Assembly.cont.capacity loop
        Buffer.cont.move(Assembly.cont);
    next;

    waituntil assembly.cont.full prio 1;
    Assembly.cont.move(exit);

    -- starts method execution again
    self.methcall(0);
end;
```

Figure 4.33 Method load

When all conditions are satisfied, the Buffer's capacity is set to match the container's capacity and then filled with entities from the "Storage", representing the set up process.

Afterwards, all the entities in the "Buffer" are loaded onto the container (Assambly.cont) so that when this is full, the truck, that is, container and entities is move to the station "exit" and then leave the frame by the interface.

As for the containers, these are created by the source, which produces one container a day and are then moved to the “Assembly”. In order for the” load” method to work properly, the containers should only move to the Assembly when there is at least one entity in the “Storage”. Otherwise, one of the conditions of the “load” method would not be fulfilled.

For that purpose, the method “truck_frequency” is implemented as an exit control for the Source (see figure 4.34).

```
is
do
    waituntil Storage.occupied prio 3;
    @.move;
    .powder.trucks:= .powder.trucks+1;
end;
```

Figure 4.34 Method truck_frequency

❖ Road transport (Transport_DDC)

The transportation between the Dairy and the DC is modeled by a track object and characterized by its transportation time and transportation temperature.

The transportation time is 1:44:10 (1,74 hours), which means a distance of 138,89 km with a speed of 80 km/h. As for the temperature, depending on the scenario being studied will be room temperature or cooling temperature.

Likewise section 4.3.6, methods “entrance_transport” and “Tt_Transport” are used as entrance and exit controls respectively.

❖ Unloading process (unload)

The unloading process is implemented in the frame “unload” and has already been described in section 4.3.6.

4.3.9 Distribution Center (DC)

The Distribution Center (DC) frame consists of three different parts: the reception and storage of the entities, the order set up for customer 1 and the order set up for customer 2 (see figure 4.35).

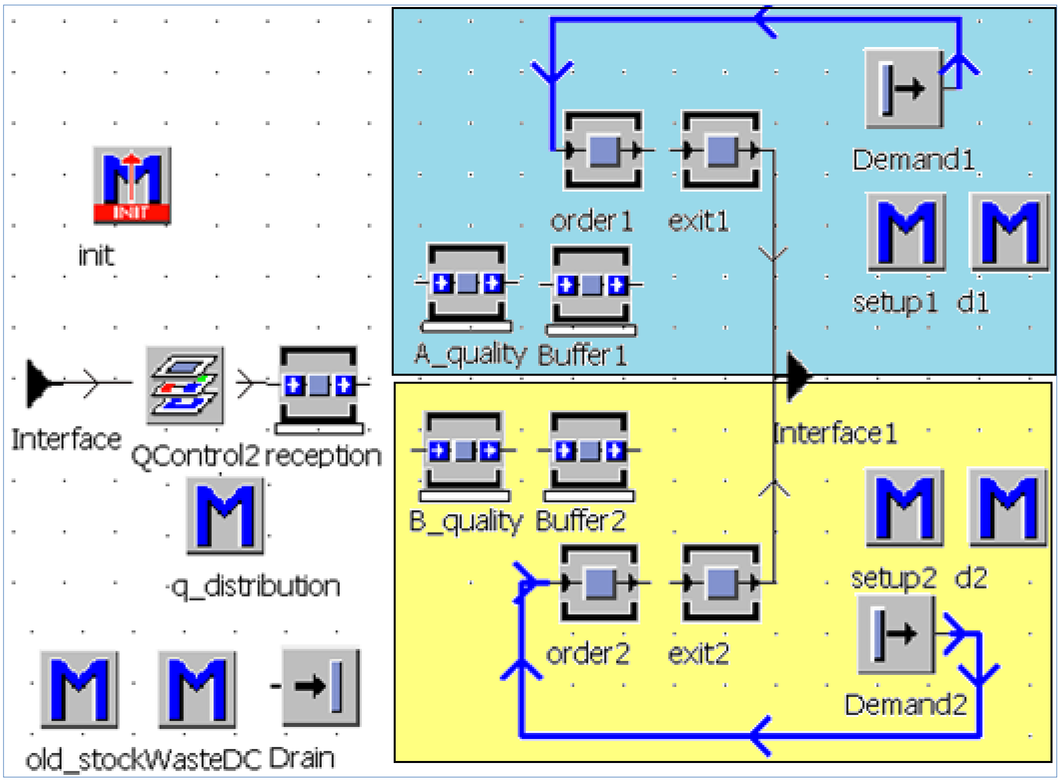

Figure 4.35 Frame DC

When the entities arrive to the DC, they firstly go through the frame “QControl”, which works in the same way as explained in section 4.3.7 but with a different quality level requirement: entities with a quality level smaller than 58 points are removed from the process and become waste.

Those entities which fulfill the quality requirements are moved to the “reception” buffer, where they will be classified by the method “q_distribution” (see figure 4.36) according to their quality: entities with a quality level greater or equal to 63 points are moved to buffer “A_quality” and the reminding entities are moved to buffer “B_quality”.

```
is
do
    if @.Quality >= 63 then
        @.move(A_quality);
    else @.move(B_quality);
    end;

end;
```

Figure 4.36 Method q_distribution

Once the entities are stored at the respective buffers, two parallel processes run for customer 1 (blue in figure 4.33) and customer 2 (yellow in figure 4.33).

- Customer 1

The DC objects modeling the distribution for customer 1 are shown in figure 4.37:

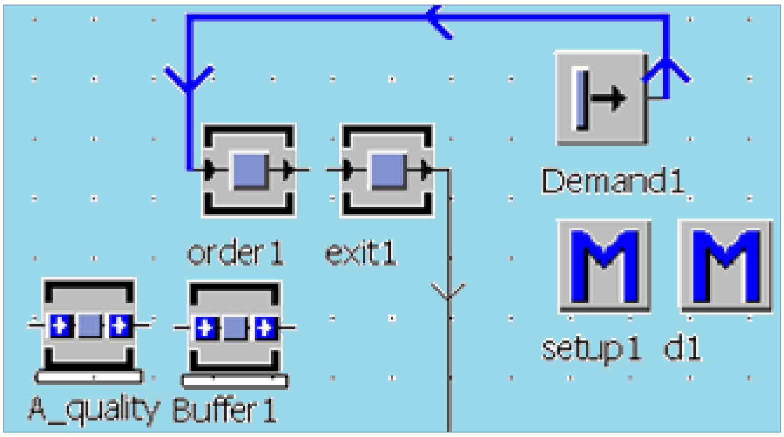

Figure 4.37 Frame DC: Customer 1

Entities with appropriate quality level for customer 1 are stored in "A_quality" until an order is placed.

The source "Demand1" starts the information flow by processing the customer's orders. This is modeled by a truck "container_c" arriving to "order1". Customer 1 uses a periodic review system and places one order a week and the order quantity is variable.

The order quantity is modeled by the container's attribute "K", and is set by the source's exit control "d1", which modifies it according to a normal distribution with seed number 2, mean 12 and standard deviation 1 (see figure 4.38).

```
is
do
    @.K:= abs(z_normal(2,12,1));
    @.move;
end;
```

Figure 4.38 Method d1

The rest of the dynamics are managed by the method “setup1” (see figure 4.39), a continuously running method initiated by the method init.

```
is
do
    -- wait until an order is recieved
    Waituntil order1.occupied
    and order1.cont.empty prio 4;
    Buffer1.capacity:=order1.cont.K;

    -- set up order quantity
    waituntil A_quality.numMU >= Buffer1.capacity
    and order1.occupied prio 3;
    for local i := 1 to Buffer1.capacity loop
        A_quality.cont.move(Buffer1);
    next;

    -- load order into truck
    for local i := 1 to Buffer1.capacity loop
        Buffer1.cont.move(order1.cont);
    next;
    order1.cont.customer:= "A";
    order1.cont.move(exit1);

    -- starts method execution again
    self.methcall(0);
end;
```

Figure 4.39 Method setup1

The method “setup1” holds the entities at “A_quality” until an empty container arrives to “order1” to then set Buffer1’s capacity to match the order quantity (container’s attribute K).

Once the necessary number of entities is available at “A_quality”, these are moved to “Buffer1” until full capacity is reached; that is, the order quantity is now available and set up.

Immediately afterwards, the prepared entities (Buffer1.cont.) are loaded onto the container waiting in order1 (order1.cont), which is then tagged by setting quality “A” to the attribute “customer”.

Finally, the truck (container and entities) leave the DC through the object “exit1” and the interface.

- Customer 2

Customer 2 operates in a very similar way to costumer 1, that is with a periodic review model: orders are placed every six days and the order quantity is set by the method “d2” likewise method “d1” but with mean 15 and standard deviation 1,5.

Since customer 2 has lower quality requirements, when there are not enough entities available at “B_quality” it is also possible to serve entities from “A_quality”. This is also implemented in method “setup2”, as shown in figure 4.40, adding some extra commands in comparison to “setup1”.

The initial conditions to start the method are the maintained, as well as the loading from “Buffer2” onto the container waiting in “order2”, nevertheless the filling of “Buffer2” is implemented differently.

```
-- set up order quantity
-- order Q available from B quality
waituntil B_quality.numMU>=Buffer2.capacity
or (B_quality.numMU+A_quality.numMU>=Buffer2.capacity
and order1.empty) prio 4;
if B_quality.numMU>=Buffer2.capacity then
    for local i := 1 to Buffer2.capacity loop
    B_quality.cont.move(Buffer2);
    next;
-- order Q not available from B quality only
else
    -- no orders for A and order Q available from A+B
        if B_quality.numMU>0 then
            for local i:=1 to B_quality.numMU loop
            B_quality.cont.move(Buffer2);
            next;
            for local j:=1 to Buffer2.(capacity-numMU) loop
                A_quality.cont.move(Buffer2);
            next;
        else
            for local i:=1 to Buffer2.capacity loop
                A_quality.cont.move(Buffer2);
            next;
        end;
end;
```

Figure 4.40 Method setup2 (abstract)

The main difference relays in the origin of the entities to fulfill the order request: if there are enough entities available at “B_quality” (B_quality.numMU) to fill Buffer 2, these entities are then moved from “B_Quality” to “Buffer2” until the maximum capacity is reached.

Otherwise, in the case that there are not enough entities at “B_quality” and there is no order to be served to customer1, also entities from “A_quality” will be used. An overview in form of a diagram is shown in figure 4.41:

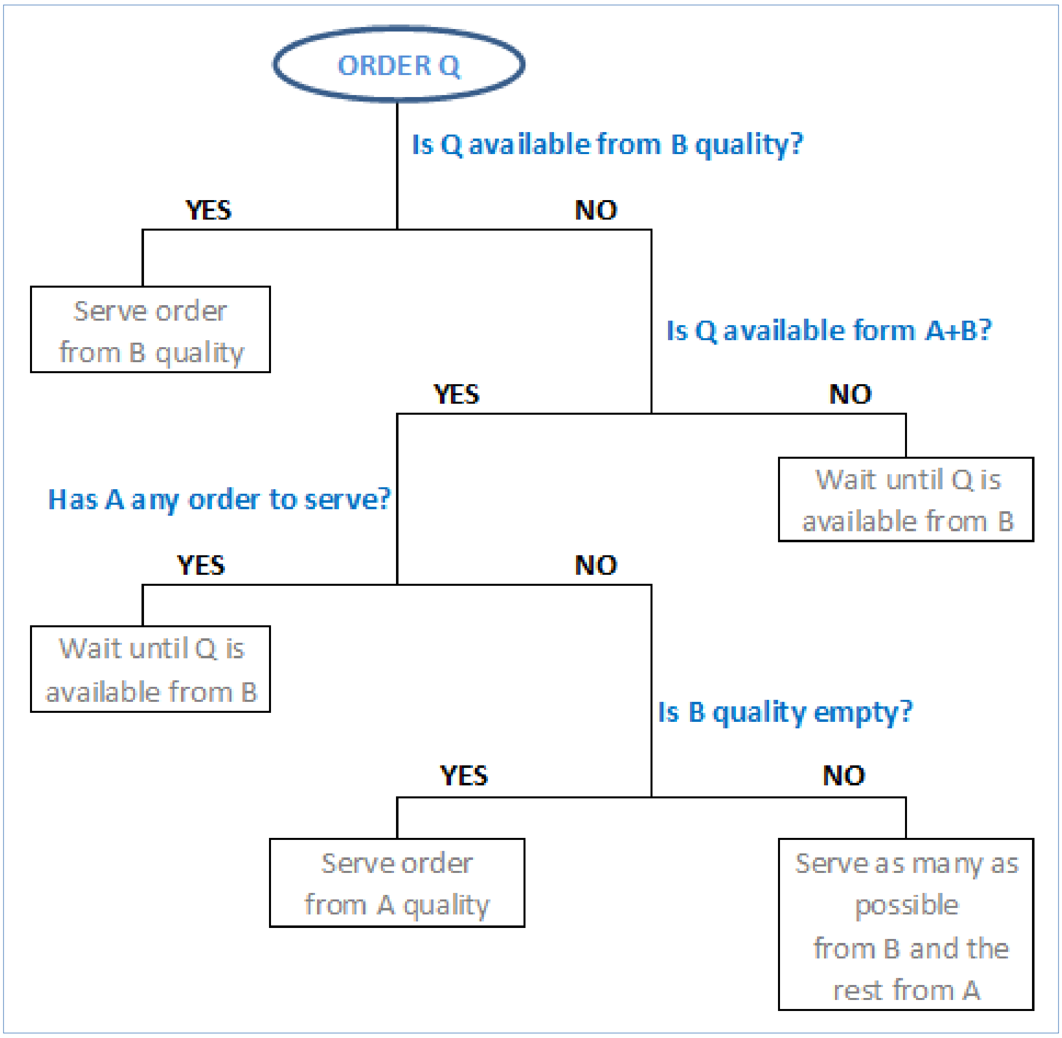

Figure 4.41 Diagram for method setup2

- Quality change

Because entities can be stored for a long time at the DC, the method “old_stock” is implemented in order to eliminate entities which no longer fulfill the quality requirements (see figure 4.42).

When either “A_quality” or “B_quality” are full, the oldest entities (FIFO stragegy), which are the most likely to have suffered quality decay, are checked: if A quality entities do no longer have a quality level grater then 61.5 points but still above 58, these are moved to “B_quality”; otherwise they are removed from the system by the Drain, as well as B quality entities with quality level below 58 are removed from “B_quality”.

```
is
do
    waituntil A_quality.full or B_quality.full prio 1;
    if A_quality.full then
        for local i := 1 to A_quality.numMU loop
            if A_quality.cont.quality<61.5 then
                if A_quality.cont.quality>=58 then
                    if B_quality.full then
                        for local j := 1 to B_quality.numMU loop
                            if B_quality.cont.quality<58 then
                                B_quality.cont.move(drain);
                            end;
                        next;
                    else
                        A_quality.cont.move(drain);
                    end;
                    A_quality.cont.move(B_quality);
                else
                    A_quality.cont.move(drain);
                end;
            end;
        next;
    end;
    if B_quality.full then
        for local j := 1 to B_quality.numMU loop
                if B_quality.cont.quality<58 then
                B_quality.cont.move(drain);
            end;
        next;
    end;
    -- starts method execution again
    self.methcall(0);
end;
```

Figure 4.42 Method old_stock

In addition, the method “old_stock” helps to prevent the software from collapsing because the number of objects exceeded 1000 as foreseen in the educational license.

Orders, that is loaded containers with its respective “customer” attribute exit the frame DC by the interface and move forward to the transportation frame.

The same structure of the frame is used for scenarios S1 to S4 except for the demand times, which are shorter for concentrates because of following assumption: dairy product manufacturers need a determined quantity of milk’s dry matter to produce the finished products, independently of the water content of the semi-finished product (powders or concentrates).

Hence, since concentrates have a much higher water content than powders, dairy manufacturers will need a higher quantity of 250kg bags to keep the same production level. For that reason, the periodic system review for both customers still apply, but the order interval is reduced to 4 days for customer 1 and 3 days for customer 2.

4.3.10 Transport from DC to customer

The transport frame is used one last time to model the transportation from the DC to the final customer, even though in this case the loading process is not included here but in the DC frame (see figure 4.43).

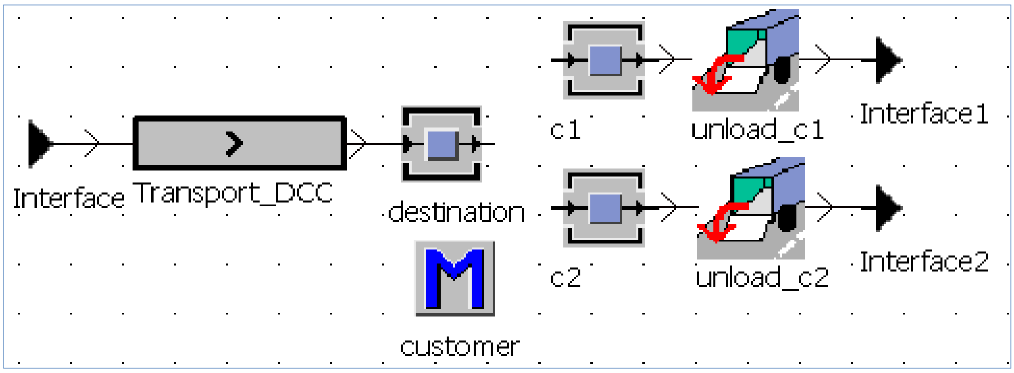

Figure 4.43 Frame Transport2

The transport is modeled by a track object and characterized by its transportation time and transportation temperature.

The transportation time is 2:46:40 (2,78 hours), which means a distance of 222,22 km with a speed of 80 km/h. As for the temperature, depending on the scenario being studied will be room temperature or cooling temperature.

Likewise section 4.3.6, methods “entrance_transport” and “Tt_Transport” are used as entrance and exit controls respectively.

Once the trucks are at the end of the truck object, they are passed on to the SingleProc “destination”, which does not represent any real process but triggers the exit control “customer”, which will read the container’s attribute “customer” and move the trucks to c1 or c2 respectively (see figure 4.44).

```
is
do
    if @.customer="A" then
        @.move(c1);
    else @.move(c2);
    end;

end;
```

Figure 4.44 Method customer

Afterwards, the containers enter the frame unload and continue the process as explained in section 4.3.6 to finally leave the simulation model by the drains “Customer1” or “Customer2”.

4.3.11 Global variables

In order to quantitatively compare the different scenarios, some general variables are introduced in the model to obtain general data from every calculation run (see table 4.9).

Table 4.9 Global variables: SC

Name	Description
Batches	Number of batches delivered to the customers (number of 250 kg entities)
Trucks	Total number of trucks used in all the transport frames
TQ	Total quality (addition of the entity's quality attribute) when delivered to the customer
kgWaste	Total weight of waste at the quality controls in kg
kgWasteDC	Total weight of waste generated at the DC in kg

These will be used to calculate the defined KPI, such as mean quality; and are implemented in several methods: “batches” and “TQ” are implemented in the method “batch_TQ” as an entrance control at the drains “customer1” and “customer2”; the global variable “trucks” is updated every time the method “load” is executed, that is, every time a container is created in the system; and finally, “kgWaste” and “kgWasteDC” are implemented in the method “Waste”, which is used as an entrance control for the drains “Waste” in the “QControl” frames and in Drain at the DC.

In addition, the working times of every process step (see table 4.10) is also calculated by the method “energy”, partially shown in figure 4.45.

Table 4.10 Global variables: time

Name	Processes involved	Description
chilling	Chilling (Farm, Farm1, Farm2, BigFarm, BigFarm1)	Computation of cooling time for the chilling process
Ctrans	Transport_FD (1 to 5)	Time of refrigerated transportation
production	SingleProcs in frame Dairy (names depend on the scenario)	Production energy consumption time
Trans	Transport_DDC	Time of refrigerated transportation in the Si.jd .COLD scenarios
storage	A_quality, B_quality (frame DC)	Time of refrigerated storage in the Si.jd.COLD scenarios
trans1	Transport_DCC	Time of refrigerated transportation in the Si.jd.COLD scenarios

```
is
do
    --Farms
    .powder.chilling:=.powder.chilling
    +.powder.CHAIN.Farm.chilling.statWorkingTime
    +.powder.CHAIN.Farm1.chilling.statWorkingTime
    +.powder.CHAIN.Farm2.chilling.statWorkingTime
    +.powder.CHAIN.BigFarm.chilling.statWorkingTime
    +.powder.CHAIN.BigFarm1.chilling.statWorkingTime;

    --Transport Farm to Dairy
    .powder.Ctrans:=.powder.Ctrans
    +.powder.CHAIN.Transport.Transport_FD1.statWorkingTime
    +.powder.CHAIN.Transport.Transport_FD2.statWorkingTime
    +.powder.CHAIN.Transport.Transport_FD3.statWorkingTime
    +.powder.CHAIN.Transport.Transport_FD4.statWorkingTime
    +.powder.CHAIN.Transport.Transport_FD5.statWorkingTime;
```

Figure 4.45 Method energy

The method “energy” is called by the method init when the Event Controller has reached the simulation time horizon, that is after 1460 simulation days.

4.4 Model verification and validation

4.4.1 Verification

Law and Kelton (2000) define verification as the determination of the model correctness, that is, whether the model has been correctly implemented in the simulation software, i.e. debugging the program.

The authors present several methods for the verification of simulation computer programs:

- Write and debug the computer program in modules or subprograms (technique1): The use of frames included in a top-down approach allowed to test each frame individually before adding it to the general model.

- Trace the state of the simulated system, i.e., the contents of the event list, the state variables, etc. and compare them to hand calculations (technique 4): because of the model complexity, a simulation run for e for only one entity was performed(see figure 4.46), in order to be able to calculate the state of the system by hand (see table 4.11).

Table 4.11 Hand calculations for a single entity

		start	finish
Farm	cows	0:00:00	0:00:00
	chilling	0:00:00	1:00:00
Transport	Buffer	1:00:00	1:20:00
	SingleProc1	1:20:00	1:20:00.1
	Assembly	1:20:00.1	1:20:00.2
	Transport_FD1	1:20:00.2	4:40:00.2
	SingleProc	4:40:00.2	5:10:00.2
	Buffer	5:10:00.2	5:10:00.2
Dairy	Buffer	5:10:00.2	5:10:00.3
	ok	5:10:00.3	5:10:00.4
	Standarization	5:10:00.4	5:31:00.4
	pasteurization	5:31:00.4	5:51:00.4
	evaporation	5:51:00.4	6:03:00.4
	spray drying	6:03:00.4	6:06:00.4
	packaging	6:06:00.4	6:08:00.4
	Buffer	6:08:00.4	6:08:00.5
	packs	6:08:00.5	6:08:00.6
	dock	6:08:00.6	6:08:00.7
Transport1	Storage	6:08:00.7	6:09:00.7
	Assembly	6:09:00.7	6:09:00.8
	Exit	6:09:00.8	6:09:01.1
	Transport_DDC	6:09:01.1	7:53:11.1
	SingleProc	7:53:11.1	8:23:11.1
	Buffer	8:23:11.1	8:23:11.1
DC	Buffer	8:23:11.1	8:23:11.2
	ok	8:23:11.2	8:23:11.3
	Reception	8:23:11.3	8:23:11.6
	A_quality	8:23:11.6	3:00:00:00.0
	Buffer1	3:00:00:00.0	3:00:01:00.1
	Order1	3:00:01:00.1	3:00:01:00.2
	Exit1	3:00:01:00.2	3:00:01:00.3
Transport2	Transport_DCC	3:00:01:00.3	3:02:46:40.3
	Destination	3:02:46:40.3	3:02:46:40.4
	c1	3:02:46:40.4	3:02:46:40.5
	SingleProc	3:02:46:40.5	3:03:16:40.5
	Buffer	3:03:16:40.5	**3:03:16:40.6**

	string 1	string 2	string 3	real 4	real 5
1	chilling	0.0000	1:00:27.8874	4.00	99.46
2	Buffer	1:00:27.8874	1:20:27.8874	4.00	99.30
3	transport	1:51:21.0304	5:11:21.0304	4.51	97.35
4	standarization	5:11:21.2304	5:32:21.2304	55.00	74.33
5	pasteurization	5:32:21.2304	5:52:21.2304	85.00	74.16
6	evaporation	5:52:21.2304	6:04:21.2304	70.00	74.13
7	spray_drying	6:04:21.2304	6:07:21.2304	195.00	65.80
8	packaging	6:07:21.2304	6:09:21.2304	28.00	65.80
9	transport	6:39:21.8304	8:23:31.8304	20.54	65.79
10	A_quality	8:23:32.3304	2:08:23:48.1363	20.00	65.79
11	transport	3:00:30:00.3000	3:03:16:40.3000	20.46	65.79

Figure 4.46 Trace of the single entity's simulation

The simulation program ends at time 3:03:16:40.3 while the calculations end at time 3:03:16:40.6. The error at the simulation end is 0,3 seconds which is completely tolerable, even so some other small discrepancies along the processes.

The timing discrepancies are due to the stochastic failure definition of the machines, which was not contemplated in the hand calculations as well as to the auxiliary steps with processing times shorter than 1 second

For those process steps representing a real process, that is, for those suffering quality decay, also the quality level is calculated and compared to the one obtained from the software (see table 4.12).

Table 4.12 Hand calculation of the quality decay

	t	t [days]	T[ºC]	T[K]	Dq	quality
chilling	1:00:00	0,0417	4	277	0,994	99,44
Buffer	20:00	0,0033	4	277	1,000	99,40
Transport_FD1	3:20:00	0,1389	4	277	0,981	97,55
Standarization	21:00	0,0146	55	328	0,749	73,05
pasteurization	20:00	0,0139	85	358	0,998	72,87
evaporation	12:00	0,0083	70	343	1,000	72,83
spray drying	3:00	0,0021	195	468	0,882	64,25
packaging	2:00	0,0014	28	301	1,000	64,25
Transport_DDC	1:44:10	0,0723	20	293	1,000	64,24
A_quality	2:15:36:49.4	2,6506	20	293	0,998	64,12
Transport_DCC	2:46:40	0,1157	20	293	1,000	**64,12**

Even though there is a small difference between the quality level calculated by PlantSimulation (65,79) and the one calculated by hand (64,12) the error represents approximately 2,5% and is due to the fact that the method “TQSL” includes temperatures random variation in the transportation frames, fact that can easily be observed from figure 4.46, for instance the entity has a temperature of 4,51ºC when transported from the farm to the dairy instead of 4ºC.

In conclusion, despite the little discrepancies in both calculations and due to the fact that in both cases the source of error has been identified; the model is verified.

4.4.2 Validation

According to Fishman and Kiviat (1968), validation is the process of determining the accuracy of the system representation by the model in order to fulfill the particular objectives of the study.

Furthermore, Law and Kelton (2000), affirm that it would be possible to use the model to make decisions about the system when this has been validated, and add some general perspectives on validation such as the fact that a complete validation is only possible if a version of the system currently exists, or that complex system can only approximately be modeled.

Hence, the presented simulation model cannot be validated so far, since the real system for concentrates is still been investigated and some data information is still missing. Furthermore, the model should be suited to a specific SC and their needs, so that validation is not completely possible at this stage but will be in the future.

5 Results and conclusions

5.1 Results

The simulation results are presented in order from S0 to S4, including: absolute results (see tables 5.1 - 5.3, 5.5, 5.7 and 5.9), the relative results in percentages (see tables 5.4, 5.6, 5.8 and 5.10), as well as the normal graph and the effects graph from the factorial design with the parameters delivery frequency and cooling (see figures 5.1 - 5.8).

- S0 (reference scenario):

Table 5.1 Results S0.2d.RT: SC

scenario	S0.2d.RT
batches	5.813
trucks	5.149
TQ	383.232,68
Q	65,927
C1	2.774
C2	3.039
kgWaste	80.530
kgWasteDC	180.500

Table 5.2 Results S0.2d.RT: transport

transportation	time [h]
farm to dairy	8412,500
dairy to DC	2532,986
DC to customer	2109,028
Production	5756,883
Storage	46020,091

❖ S1

Table 5.3 Results S1

scenario	S1.1d.RT	S1.1d.COLD	S1.2d.RT	S1.2d.COLD
batches	10.254	11.197	10.533	9.815
trucks	8.764	8.786	5.120	5.118
TQ	605.930,71	678.798,78	623.629,88	606.112,81
Q	59,092	60,623	59,207	61,754
C1	3.218	4.178	3.426	2.745
C2	7.036	7.080	7.107	7.070
kgWaste	75.200	74.211	72.746	73.996
kgWasteDC	971.500	679.750	905.000	1.080.000
production [h]		4662,056		
storage [h]	67147,218	70017,9404	66841,14786	67181,6334

Table 5.4 Results S1 in percentage

scenario	S1.1d.RT	S1.1d.COLD	S1.2d.RT	S1.2d.COLD
batches	176%	193%	181%	169%
trucks	170%	171%	99%	99%
TQ	158%	177%	163%	158%
Q	90%	92%	90%	94%
C1	116%	151%	124%	99%
C2	232%	233%	234%	233%
kgWaste	93%	92%	90%	92%
kgWasteDC	538%	377%	501%	598%
production [h]		81%		
storage [h]	146%	152%	145%	146%

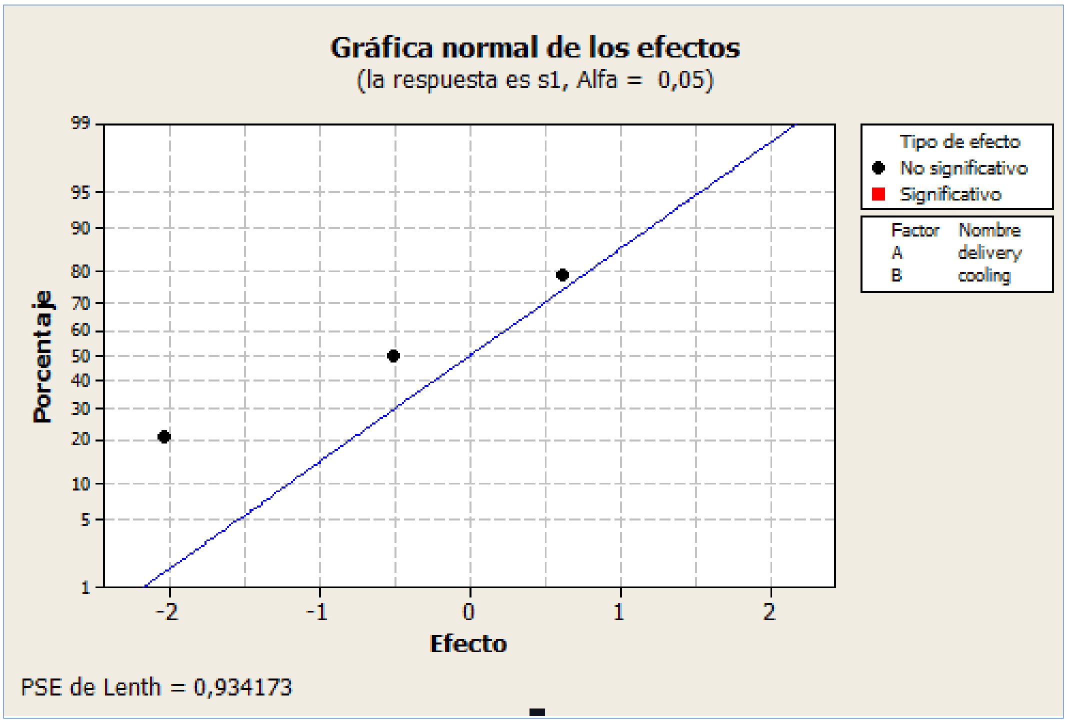

Figure 5.1 S1 effects normal graph

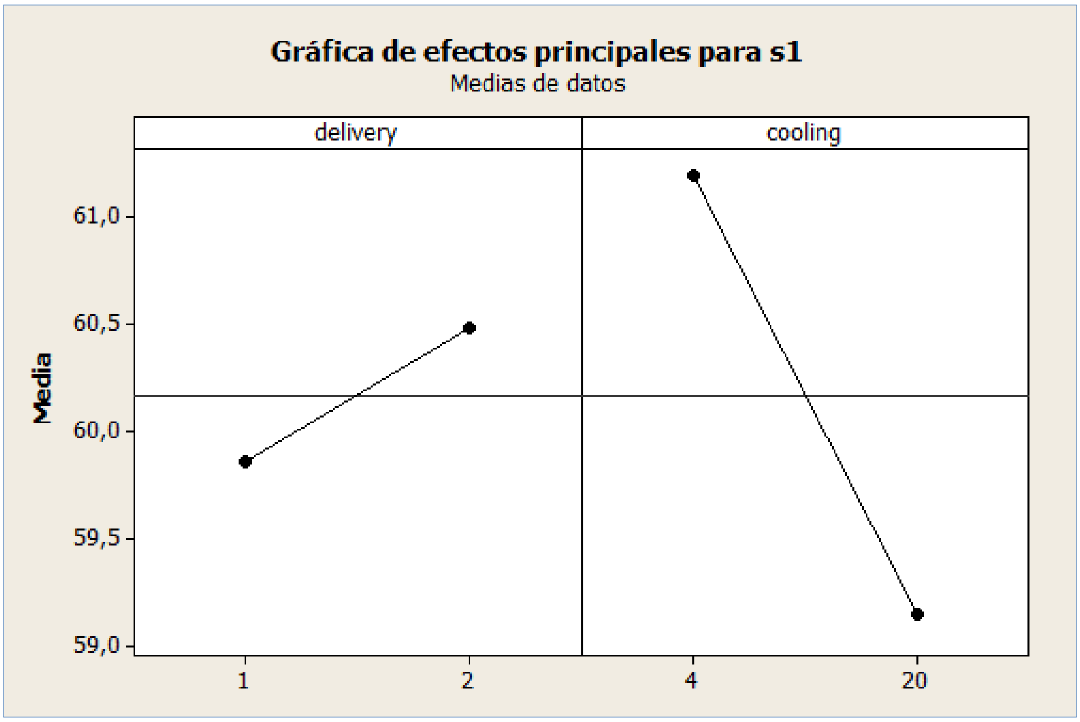

Figure 5.2 S1 effects

❖ S2

Table 5.5 Results S2

scenario	S2.1d.RT	S2.1d.COLD	S2.2d.RT	S2.2d.COLD
batches	11.197	11.208	11.229	11.209
trucks	8.769	8.766	5.114	5.111
TQ	674.402,43	713.874,01	677.631,95	714.969,17
Q	60,231	63,693	60,347	63,785
C1	4.175	4.177	4.207	4.194
C2	7.022	7.031	7.022	7.015
kgWaste	41.244	49.076	51.435	42.982
kgWasteDC	770.000	759.750	752.500	765.500
production [h]	5047,174			
storage [h]	43020,343	41234,8863	41657,73978	39678,87

Table 5.6 Results S2 in percentage

scenario	S2.1d.RT	S2.1d.COLD	S2.2d.RT	S2.2d.COLD
batches	193%	193%	193%	193%
trucks	170%	170%	99%	99%
TQ	176%	186%	177%	187%
Q	91%	97%	92%	97%
C1	151%	151%	152%	151%
C2	231%	231%	231%	231%
kgWaste	51%	61%	64%	53%
kgWasteDC	427%	421%	417%	424%
production [h]	88%			
storage [h]	93%	90%	91%	86%

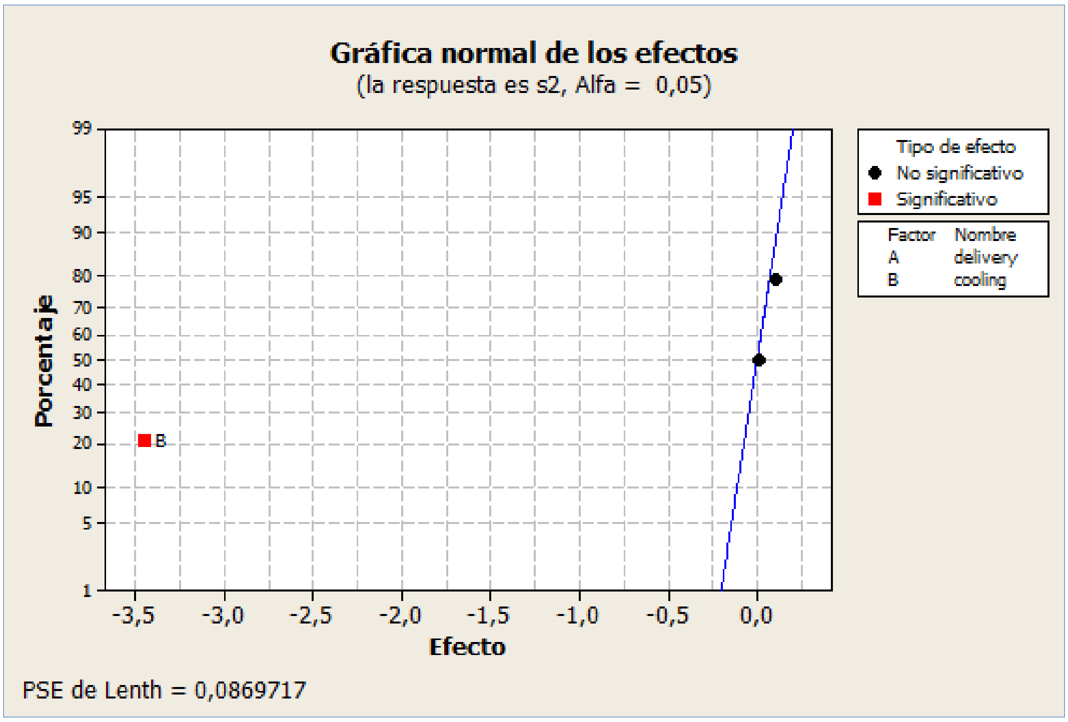

Figure 5.3 S2 effects normal graph

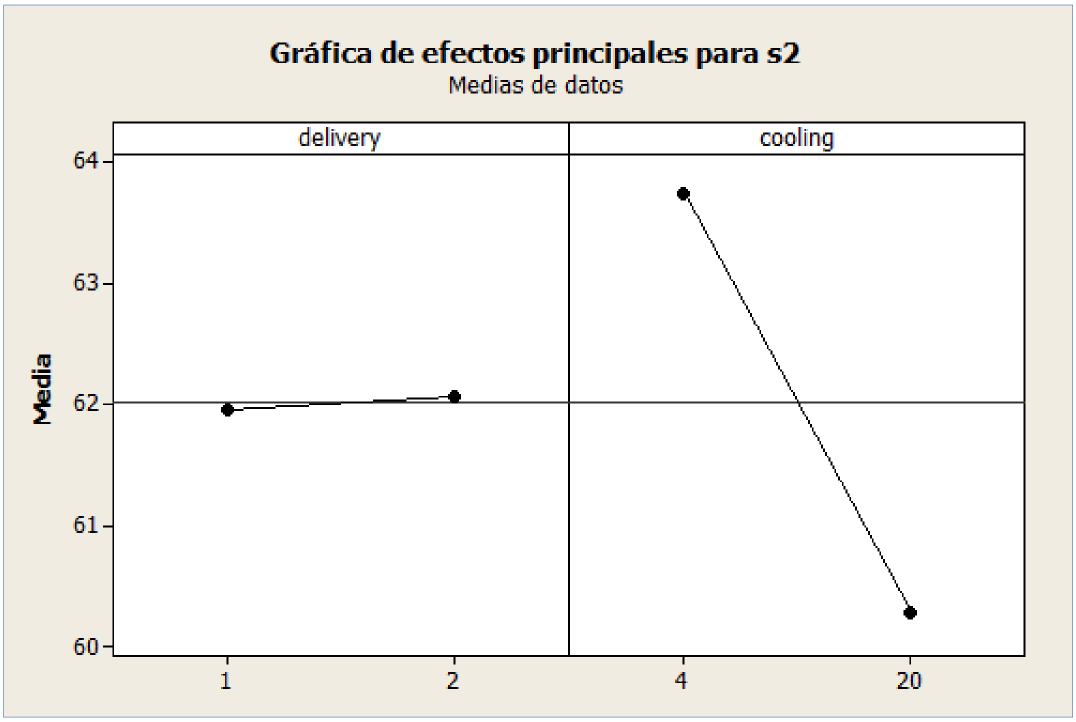

Figure 5.4 S2 Effects

- S3

Table 5.7 Results S3

scenario	S3.1d.RT	S3.1d.COLD	S3.2d.RT	S3.2d.COLD
batches	11.199	11.223	11.217	11.231
trucks	8.763	8.761	5.115	5.115
TQ	702.703,90	744.365,57	709.552,40	745.552,47
Q	62,747	66,325	63,257	66,383
C1	4.207	4.185	4.185	4.157
C2	6.992	7.038	7.032	7.074
kgWaste	39.492	40.378	43.958	44.029
kgWasteDC	771.500	764.500	762.750	758.750
production [h]	4682,000			
storage [h]	36689,584	36251,5236	35705,32622	35843,2113

Table 5.8 Results S3 in percentage

scenario	S3.1d.RT	S3.1d.COLD	S3.2d.RT	S3.2d.COLD
batches	193%	193%	193%	193%
trucks	170%	170%	99%	99%
TQ	183%	194%	185%	195%
Q	95%	101%	96%	101%
C1	152%	151%	151%	150%
C2	230%	232%	231%	233%
kgWaste	49%	50%	55%	55%
kgWasteDC	427%	424%	423%	420%
production [h]	81%			
storage [h]	80%	79%	78%	78%

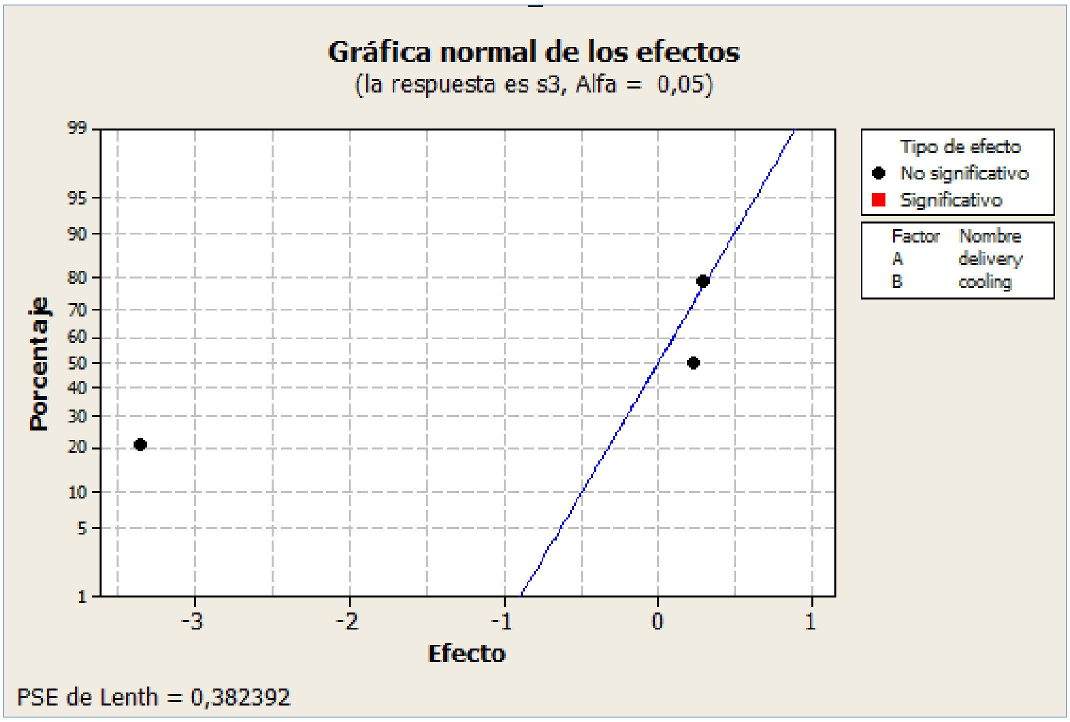

Figure 5.5 S3 effects normal graph

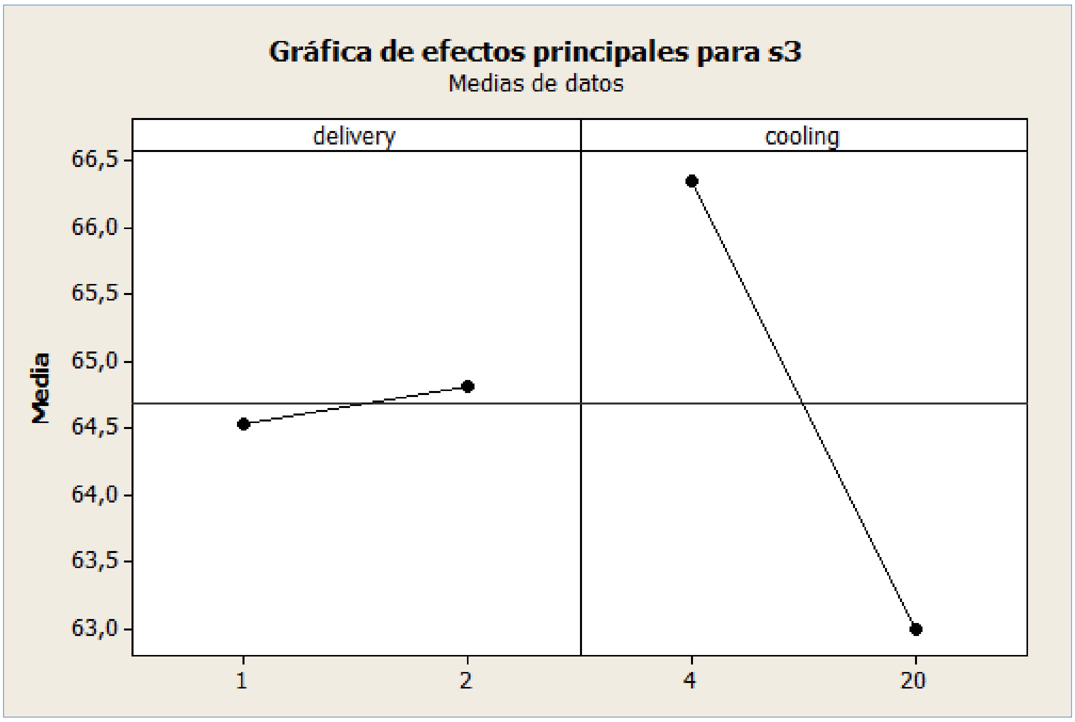

Figure 5.6 S3 effects

❖ S4

Table 5.10 Results S4

scenario	S4.1d.RT	S4.1d.COLD	S4.2d.RT	S4.2d.COLD
batches	11.215	11.195	11.241	11.250
trucks	8.764	8.766	5.112	5.116
TQ	697.949,74	736.503,95	700.260,95	740.852,82
Q	62,234	65,789	62,295	65,854
C1	4.175	4.171	4.183	4.194
C2	7.040	7.024	7.058	7.056
kgWaste	42.469	44.587	49.844	46.683
kgWasteDC	764.000	767.500	750.500	751.500
production [h]		4681,674		
storage [h]	36586,857	36342,9582	36395,00947	36319,1614

Table 5.11 Results S4 in percentage

Scenario	S4.1d.RT	S4.1d.COLD	S4.2d.RT	S4.2d.COLD
Batches	193%	193%	193%	194%
Trucks	170%	170%	99%	99%
TQ	182%	192%	183%	193%
Q	94%	100%	94%	100%
C1	151%	150%	151%	151%
C2	232%	231%	232%	232%
kgWaste	53%	55%	62%	58%
kgWasteDC	423%	425%	416%	416%
production [h]		81%		
storage [h]	80%	79%	79%	79%

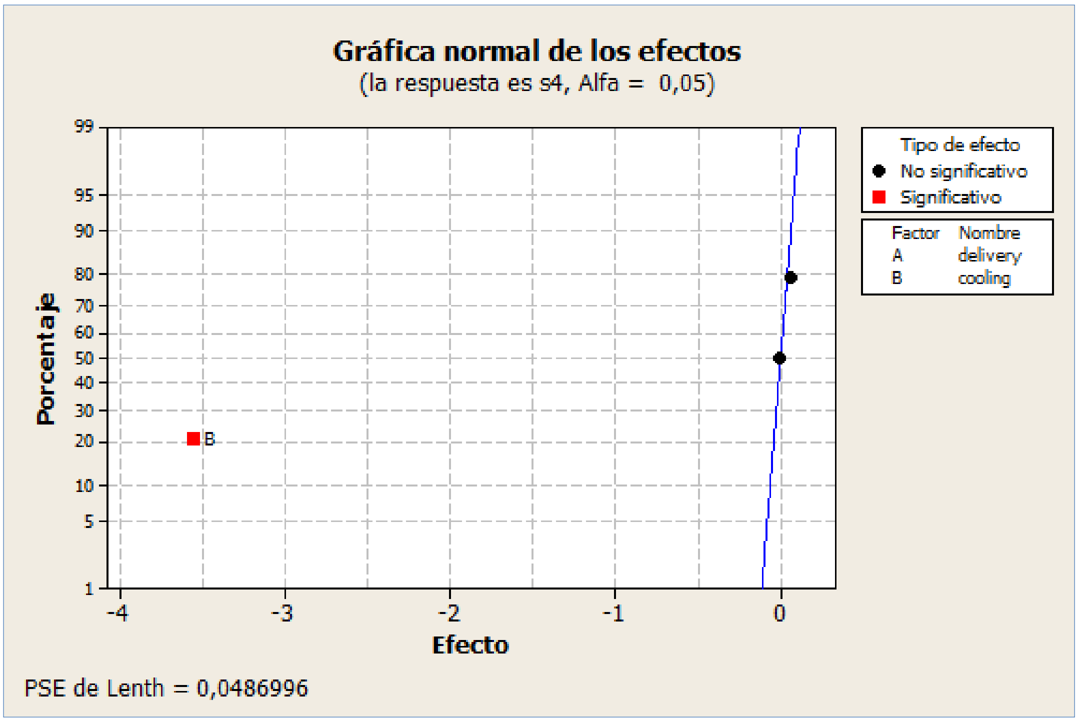

Figure 5.7 S4 effects normal graph

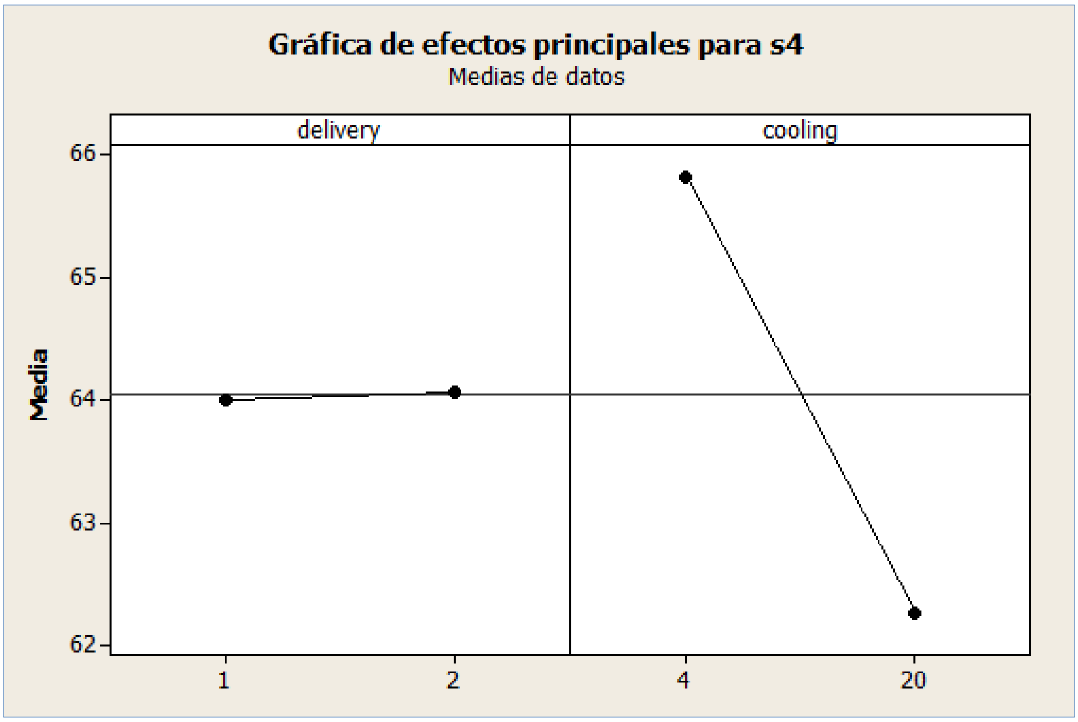

Figure 5.8 S4 effects

5.2 Results: TC analysis

The KPI Total Cost is calculated as following:

production + transport + ***cooling*** transport + storage + ***cooling*** storage + waste

For that purpose, in table 5.12 an overview of the times for each of the parameters analyzed is presented, where the results are standardized, that is, the absolute value is divided by the number of batches obtaining [process hours/batches] which is proportional by a factor of 1/250 to [process hours/kg].

Table 5.12 Standardized times

	production	transport	cooling transport	storage	cooling storage	waste	wasteDC
S0.2d.RT	0,9903	2,2457	1,4472	7,9168	0,0000	13,8534	31,0511
S1.1d.RT	0,4547	2,0935	1,6408	6,5484	0,0000	7,3337	94,7435
S1.1d.COLD	0,4164	1,9172	1,9172	6,2533	6,2533	6,6278	60,7082
S1.2d.RT	0,4426	1,2394	0,7987	6,3459	0,0000	6,9065	85,9204
S1.2d.COLD	0,4750	1,3301	1,3301	6,8448	6,8448	7,5391	110,0357
S2.1d.RT	0,4508	1,5026	1,5026	3,8421	0,0000	3,6835	68,7684
S2.1d.COLD	0,4503	1,9153	1,9153	3,6791	3,6791	4,3787	67,7864
S2.2d.RT	0,4495	1,1626	0,7492	3,7098	0,0000	4,5806	67,0140
S2.2d.COLD	0,4503	1,1646	1,1646	3,5399	3,5399	3,8346	68,2933
S3.1d.RT	0,4181	1,5024	0,7512	3,2761	0,0000	3,5264	68,8901
S3.1d.COLD	0,4172	1,9128	1,9128	3,2301	3,2301	3,5978	68,1190
S3.2d.RT	0,4174	1,1638	1,5000	3,1831	0,0000	3,9189	67,9995
S3.2d.COLD	0,4169	1,1624	1,1624	3,1915	3,1915	3,9203	67,5585
S4.1d.RT	0,4174	1,5002	0,7501	3,2623	0,0000	3,5214	68,7918
S4.1d.COLD	0,4182	1,9176	1,9176	3,2464	3,2464	3,6068	68,2894
S4.2d.RT	0,4165	1,1613	1,4968	3,2377	0,0000	3,9105	67,8543
S4.2d.COLD	0,4161	1,1604	1,1604	3,2284	3,2284	3,9137	67,4444

In order to evaluate the obtained results, more detailed information is needed, for instance, energy consumption of each of the process machines for the production times and the storage times at the DC, cooling cost during transportation, etc.

A possibility would be, as used by Zanoni and Zavanella (2011) to estimate the costs as shown in table 5.13:

Table 5.13 Cost calculation (Zaononi and Zavanella, 2011)

production cooling transport cooling storage	Transport	Storage	Waste
SEC·ec·d (A)	K·d/Q (B)	K·d/Q+Q/2·hs (C)	d/Q·p·dq (D)

Where: SEC (Specific Energy Consumption) is the energy required to produce or to refrigerate a kg [kWh/kg], ec is the cost of energy [€/kWh] and K is the set-up cost at each location [€].

It should also be taken into consideration that, under the assumption that final producers need a specific amount of dry matter per product kg, 1 kg powders (98% dry matter) is equivalent to approximately 3 kg concentrates (30% dry matter). For instance, the relative TC for S1.1d.COLD would be:

Table 5.14 Example of relative TC calculation

	production	transport	cooling transport	storage	cooling storage	waste	wasteDC
S0.2d.RT	1,0000	1,0000	1,0000	1,0000	-	1,0000	1,0000
S1.1d.COLD	0,4204	0,8537	1,3248	0,7899	6,2533	0,4784	1,9551
eqivalent S1.1d.COLD	1,2613	2,5611	3,9744	2,3696	18,7598	1,4353	5,8653

Using the equivalent relation shown in table 5.14, the relative TC comparing S1.1d.COLD to the reference scenario would be:

$$TC = (1{,}26+3{,}97+18{,}76)\cdot A + 2{,}56\cdot B + 2{,}37\cdot C + (1{,}4+5{,}87)\cdot D$$

$$TC = 24\cdot A + 2{,}56\cdot B + 2{,}37\cdot C + 7{,}3\cdot D$$

5.3 Results interpretation and conclusions

As mentioned in the previous sections, part of the data used for the simulation model is fictional, since the empirical data has not been collected yet, e.g. the fictional customers and their demands, as well as some of the processing times. For that reason, analysis and interpretation of the results should not serve for further research phases yet, but only to prove that the model outcome is reasonable.

For that purpose, and under all the assumptions made for the model and describe in chapter 4, following is observed:

- From the factorial design can be determined that the factor “delivery frequency” is not significant for any of the process variations (S1 to S4). Regarding the factor “cooling”, the results show that it is significant in scenarios S2 and S4 but not in S1 and S3.
- From the TC analysis can be concluded that transport, storage and waste increase considerably for the concentrates respect to powders, since factor B, C and D remain equal for all the scenarios. Regarding

the significant increased observed in the production and cooling energy, it is not possible to evaluate it yet, since the factor A includes the SEC, which has to be determined for concentrates yet and is substantially different from powders, fact that motivated indeed the global project. In conclusion, the objective is to determine whether the trade-off between the energy part and the logistic cost turns out positive for the new SC; i.e. for the case of scenario S1.1d.COLD, if:

$$\overbrace{A^{P} + B + C + D}^{\text{TC (S0.2d.RT)}} < \overbrace{24 \cdot A^{C} + 2{,}56 \cdot B + 2{,}37 \cdot C + 7{,}3 \cdot D}^{\text{TC (S1.1d.COLD)}}$$

$$\underbrace{|24 \cdot A^{C} - A^{P}|}_{\Delta\ \text{Energy Cost}} > \underbrace{1{,}56 \cdot B + 1{,}37 \cdot C + 6{,}3 \cdot D}_{\Delta\ \text{Logistic Costs}}$$

6 Model limitations and further research

As already mentioned through the thesis, some improvements can be made to the simulation model:

- Empirical data

In the following research stages empirical data will be available and should be incorporated to the model, as well as more detailed processing when necessary, by adding frames on the already existing layer.

Empirical data should also provide another verification possibility, i.e. the comparison between these and the simulation outcomes, which should be similar. Otherwise, the source of the discrepancy should be found.

Moreover, experts on the matter, such as the partner chair, responsible for the bio-processing and food engineering, or the industry partners should understand the simulation model and validate it after the appropriate data has been introduced.

- Modeling with full license

The model's accuracy could be improved by adding some more sophisticated processes, which requires a higher operation capacity than the provided by the educational license (1000 objects).

Thus, the continuous flow characteristic of some of the processes involved in the dairy industry could be implemented by using a method that divides entities, likewise the method “batch”.

The entities could be stored in an intermediate buffer before entering a continuous process and then be divided into the necessary amount of smaller entities, to properly characterize the continuous flow.

Furthermore, turning to the “batch” method, the waste produced at the dairy could be reduced to fit the real process by grouping the entities in smaller portions (10 kg grouping used in the method).

- Demand and service level

Accurate demand estimations could be used in the implementation of the sources at the DC; moreover, once some of the scenarios have been dismissed and the research is at a more advanced state, the stock level could be taken as a parameter to study, especially regarding the concentrates’ shelf-life. At this stage, also the study of the service level at the DC could become an interesting KPI for the simulation study.

- Other factors

Other factors considered at the beginning of this thesis that could be worth investigating in the future are the packaging materials used for concentrates, the delivery frequencies from the DC to the different costumers or the reutilization of waste at some stages for low-quality concentrates.

7 Bibliography

Akkerman , R., Van Donk, D.P. (2007), Product prioritization in a two-stage food production system whith intermediate storage, *International Journal of Production Economics,* Vol. 108, No. 1-2, pp. 43-53.

Akkerman, R., van Donk, D.P. (2010), Balancing environmental and economic performance in the food processing industry*, International Journal of Entrepreneurship and Innovation Management*, Vol. 11 , No. 3, pp.330-340.

Akkerman, R., Farahani, P., Grunow, M., (2010), Quality, safety and sustainability in food distribution: A review of quantitative operations management research*, OR Spectrum* 32 (4), pp.863-904.

Aramyan, L., Ondersteijn, C., Kooten, O., Lansink, A.O. (2006), Performance indicators in agri-food production chains*, Quantifying the agri-food Supply Chain*, Springer, pp. 49-66.

Beamon, B.M., Chen, V.C.P. (2001), Performance Analysis of conjoined Supply Chains*, International Journal of Production Research*, Vol. 39, No. 14, pp. 3195-3218.

Bell, P.C., Anderson, C.K., Staples, D. S., Elder, M. (1999), Decision-makers' perceptions of the value and impact of visual interactive modeling, *International Journal of Management Science*, Vol. 27 (2), pp. 155-156.

Blackburn, J., Scudder, G. (2009), Supply chain strategies for perishable good products: the case of fresh produce, *Production and Operations Management,* Vol.18, No. 2, pp. 129-137.

Bogataj, M., Bogataj, L., Vodopivec, R. (2005), Stability of perishable goods in cold logistic chains*, International Journal of Production Economics*, Vol. 93-94, pp. 345-356.

Bylund, G. (1995), *Dairy Processing Handbook,* TetraPak Processing Systems AB, Lund (Sweden), Chapter 1-5.

Den Ouden, M., Dijkhuizen, A.A., Ruud, B.M., Huirne, P., Zuurbier, J.P. (1996), Vertical cooperation in agricultural production-marketing chains, with special reference to product differentiation in pork, *Agribusiness*, Vol.12(3), pp. 277–290.

Fishman, G.S., Kiviat, P.G. (1968), The statistics of discrete-event simulation, *SIMULATION,* Vol. 10, No. 4,pp. 185-195.

Fortuin, L. (1988), Performance indicators: Why, where and how?, *European Journal of Operational Research*, Vol.34 (1), pp. 1-9.

Fu, B., Labuza, T.P. (1993), Shelf-life prediction: theory and application, *Food Control*, Vol. 4 (3),pp. 125-133.

Fu, B., Taoukis, P.S., Labuza, T.P. (1991), Predictive Microbiology for Monitoring Spoilage of Dairy Products with Time-temperature Integrators, *Journal of Food Science,* Vol. 56, No. 5.

Grunert, G.G. (2005), Food quality and safety: consumer perception and demand, *European Review of Agricultural Economics*, Oxford,pp. 369-391.

Huang, G.Q., Lau, J.S.K., mak, K.L. (2003), The impacts of sharing production information on supply chain dynamics: a review of the literature. *International Journal of Production Research*, Vol. 41 (7), pp.1483-1517.

Kessler, H. (2002), Food and bio process engineering: dairy technology, 5th ed., Verlag A. Kessler, University of Michigan.

Kleijnen, J.P.C., Smits, M.T. (2003), Performance metrics in supply chain management. *Journal of the Operational Research Society*, Vol.54 (5), pp.507-514.

Kulozik, U., Grunow, M. (2011), BMELV-Vorhaben: Energieeinsparung durch haltbare Milch- und Molkenkonzentrate.

Law, A.M., Kelton, W.D. (2000), *Simulation Modeling and Analysis,* 3rd ed., McGraw-Hill, chapter 5.

Lohman, C., Fortuin, L., Wouters, M. (2004), Designing a performance measurement system: A case study*, European Journal of Operational Research*, Vol. 156 (2), pp.267-286.

Luning, A.L., Marcelis, J.M. (2006), A techno-managerial approach in food quality management research, Trends in Food Science & Technology, Vol. 17 (7), pp. 378-385.

Manikas, I., Manos, B. (2008), Design of an integrated supply chain model for supporting traceability of dairy products*, International Journal of Dairy Technology,* Vol.62 (1), 126-138.

McHaney, R., Cronan,T.P. (1998), Computer simulations success: on the use of the end-user computing satisfaction instrument: a comment*, Decision Sciences,* Vol. 29(2), pp. 525-536.

Montanari, R. (2008), cold chain tracking: a managerial perspective, *Trends in Food Science & Technology,* Vol. 19 (8), pp. 425-431.

Quariguasi, J., Bloemhof-Ruwaard, J.M., van Nunen, J.A.E.E., van Heck, E. (2008), Designing and evaluating sustainable logistics networks*, International Journal of Production Economics*, Vol. 111(2), pp. 195-208.

Robinson, S. (2002), General concepts of quality for discrete-event simulation*, European Journal of Operational Research*, Vol.138 (1), 103-117.

Rong, A., Akkerman, R., Grunow, M. (2010): An optimization approach for managing fresh food quality throughout the supply chain*, International Journal of Production Economics*, Vol. 131, pp.421-429.

Rutten, W.G.M.M. (1995), The use of recipe flexibility in production planning and inventory control, Dissertation Eindohoven University of Technology, The Netherlands.

Schouten, R.E., tijskens, L.M.M., Van kooten, O. (2002), Predicting keeping quality of batches of cucumber fruit based on a physiological mechanism, *Postharvest Biology and Technology*, Vol.26 (2), pp. 35-47.

Subramanian, R., Talbot, B., Gupta, S. (2010), An approach to integrating environmental considerations within managerial decision-making, *Journal of Industrial Ecology*, Vol. 14 (3), pp. 378-398.

Taoukis, P.S., Labuza, T.P. (1999), Applicability of time temperature indicators as shelf life monitors of food products, *Journal of Food Science*, Vol. 54 (4), pp. 783-788.

Tecnomatix Plant Simulation Help, Plant Simulation version 10.

Tijskens, P. (2004), Discovering the future, modeling quality matters, Wageningen University, Netherlands.

Trienekens, J., Zuurbier, P. (2008), Quality and safety standards in the food intustry, developments and challenges, *International Journal of Production Economics*, Vol. 113 (1), pp. 107-122.

Van der Vorst, J.G.A.J. (2000), Effective Food Supply Chains: Generating, Modelling and Evaluating Supply Chain Scenarios, PhD-thesis Wageningen University.

Van der Vorst, J.G.A.J., Beulens, A J.M. (2002), Identifying sources of uncertainty to generate supply chain redesign strategies, *International Journal of Physical Distribution & Logistics Management*, Vol. 32 Iss: 6, pp.409 – 430.

Van der Vorst, J., Beulens, A., van Beek, P. (1999), Modelling and simulating multi-echelon food systems, *European Journal of Operational Research*, Vol.122, pp. 354-366.

Van der Vorst, J.G.A.J., Beulens, A.J.M, Van Beek, P. (2005), Innovations in logistics and ICT in food supply chain networks, in eds. Innovation in agri-food systems, product quality and consumer acceptance*,* Wageningen Academic publishers, pp. 245-292.

Van der Vorst, J., Tromp, S., van der Zee, D. (2008): Simulation modeling for food supply chain redesign; integrated decision making on product quality, sustainability and logistics, *International Journal of Production Research*, Vol. 47, No.23.

Van der Vorst, J.G.A.J., Tromp, S.O., van der Zee, D.J. (2009), Simulation modelling for sustainability and logistics, *International Journal of Production Research,* Vol. 47 (23),pp. 6611-6631.

Van der Zee, D.J., Van der Vorst, J.G.A.J. (2005), A modelling framework for supply chain simulation – opportunities for improved decision making*, Decision Sciences*, Vol.36 (1), pp. 65-95.

Van Donselaar, K., van Woensel, T., Broekmeulen, R., Fransoo, J. (2006), Inventory control of perishables in supermarkets*, International Journal of Production Economics,* Vol. 104 (2), pp. 462-472.

Van Rijn, Th.M.J., Schijns, B.V.P. (1993), the applicability of MR-II in the semi-process industry, *Assen: Van Gorcum.*

Wang, Y., Birkved, M., Akkerman, R., Grunow, M. (2011), Combined supply chain planning and sustainability assessment for professionally prepared meals, *CIRP Annals-Manufacturing .Technology*

Zanoni, S., Zavanella, L., (2011), Chilled or frozen? Decision strategies for sustainable food supply chains, *International journal of Production Economics.*